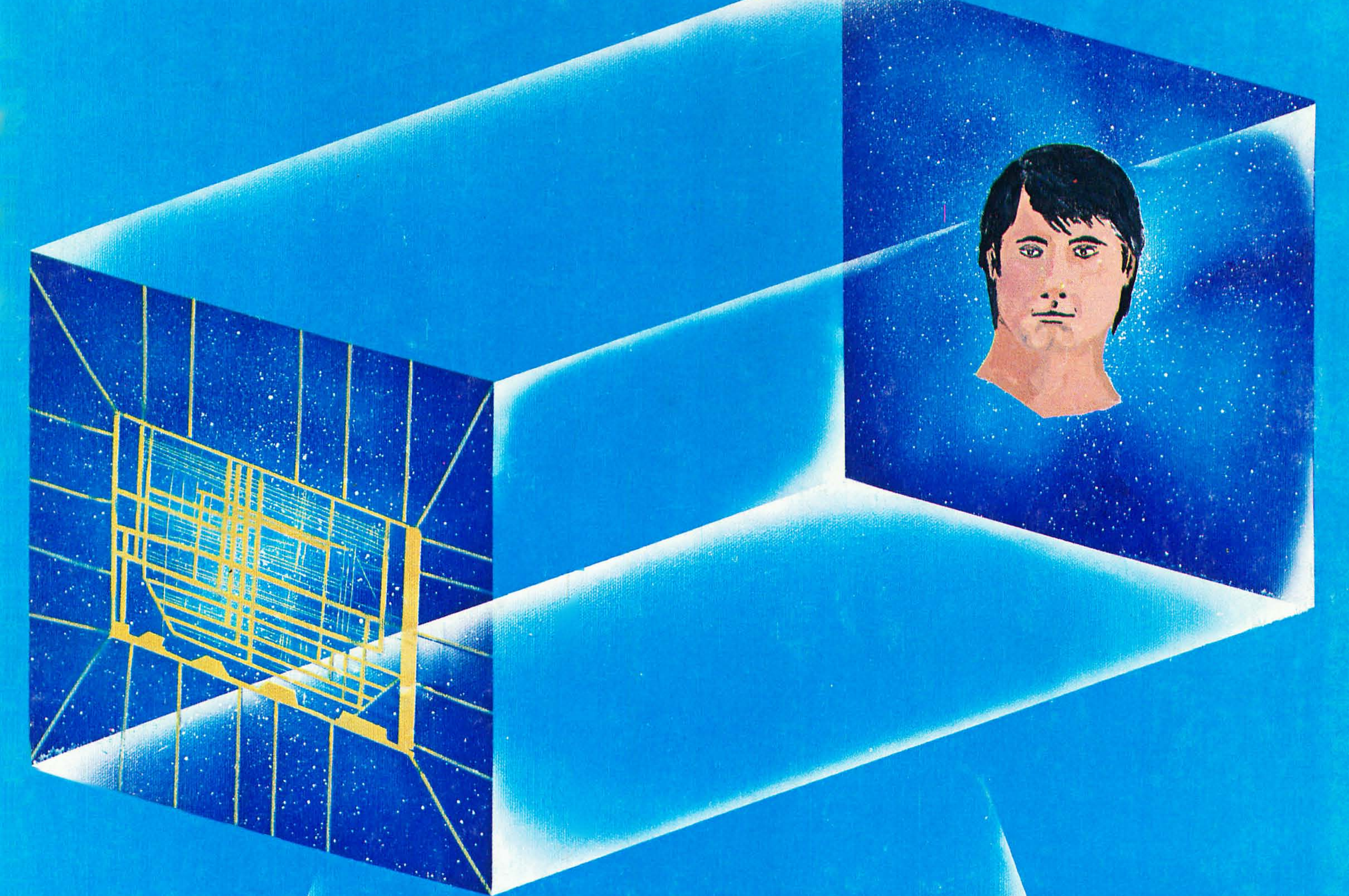
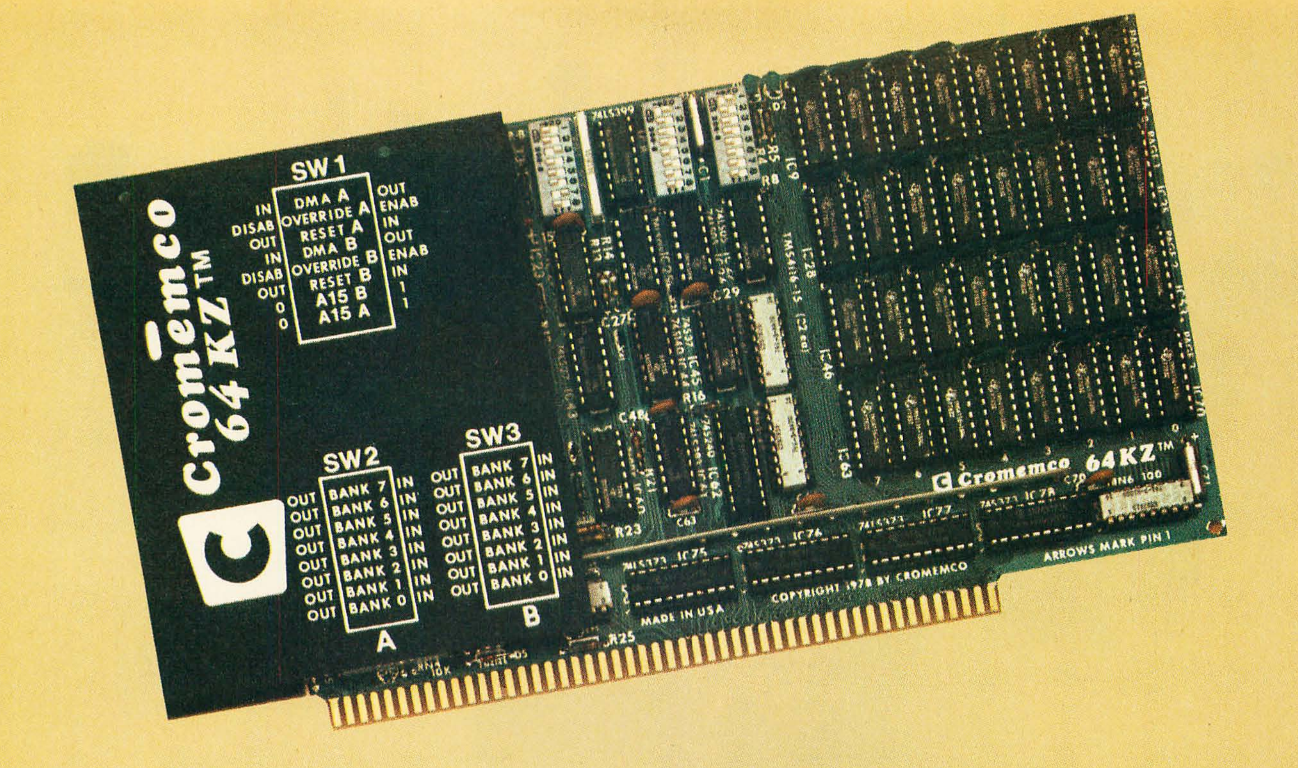
JUNE 1979 Volume 4, Number 6 $2.00 in USA/$2.40 in Canada

BYTE

the small systems journal





You can do surprising things when you have 64 kilobytes of fast RAM on one card

4 MHz FAST—AND EXPANDABLE

Here's 64 kilobytes of memory on one RAM card. Yes, we mean 512K bits of read/write memory on this single card.

And, yes, we mean it's fast. With 150-nanosecond chip access times — so the card can operate in fast Z-80 systems with no wait states. Repeat, no wait states.

EXPANDABLE ON TWO LEVELS

Not only does the new Model 64KZ give you a large, fast RAM but it is expandable on two levels.

First, through our Cromemco Bank Select feature, you can expand to 512 kilobytes in eight 64K banks.

Or, with our Extended Bank Select feature, you can expand memory space to as much as 16 megabytes.

This expandability we call your obsolescence insurance.

The legend on the card's heat sink is an easy reference for address and bank selection.

BENCHMARK IT

Obviously, the speed and memory capacity of this new card give you a lot of power.

You can see that for yourself in our new 7-station Multi-User Computer System which uses these Model 64KZ cards. This S100-bus system outperforms the speed of many if not most timesharing systems of up to 10 times the Cromemco price.

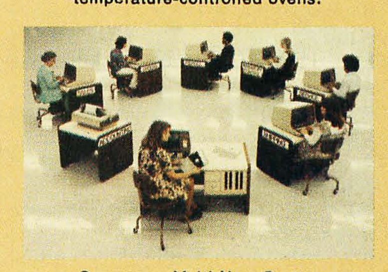
And yet where some of these much more expensive and cumbersome systems clearly slow to a snail's pace when timesharing, the Cromemco system using Bank Select switching runs surprisingly fast.

SEE IT NOW

See the new Model 64KZ at your computer dealer now. Study the literature on it. See how for only $1785 you can get around that ever-present barrier of memory that's too little and too slow.



For high reliability all Cromemco memory cards are burned in at the factory in these temperature-controlled ovens.



Cromemco Multi-User System shown with 7 stations



Cromemco
incorporated
280 BERNARDO AVE., MOUNTAIN VIEW, CA 94040 • (415) 964-7400
Tomorrow's computers now

Circle 87 on inquiry card.

Here's how you can be fully computerized for so much less than you thought

BUSINESS — EDUCATION — ENGINEERING — MANUFACTURING

We are pleased to announce the first professional time-sharing system in the microcomputer field.

Naturally, it's from Cromemco.

This new multi-user system will do all of the tasks you usually associate with much more expensive time-sharing computers. Yet it's priced at an almost unbelievably low figure.

Look at these features:

- **You can have up to 7 terminals plus a fast, 132-column line printer**
- **You can have a large system RAM memory that's expandable to ½ megabyte using the Bank Select feature**
- **Each user has an independent bank of RAM**
- **You can have floppy disk storage of up to 1 megabyte**
- **You have confidentiality between most stations**
- **And, make no mistake, the system is fast and powerful. You'll want to try its fast execution time yourself.**



PROGRAMMERS LOVE OUR BASIC

This new system is based on Cromemco's well-known System Three Computer and our new Multi-User BASIC software package.

Programmers tell us that Cromemco Multi-User BASIC is the best in the field. Here are some of its attractions:

- **You can use long variable names and labels up to 31 characters long — names like "material on order" or "calculate speed reduction."**
- **You get many unusual and helpful commands that simplify programs and execution — commands such as PROTECT, LIST VARIABLES, NOLIST, and many more.**
- **No round-off error in financial work (because our BASIC uses binary-coded decimal rather than binary operation). And we've still been able to make it FAST.**
- **Terminals and printer are interrupt-driven — no additional overhead until key is pressed.**
- **The conveniences in this Multi-User BASIC make it much easier to write your own application software.**
- **A line editor simplifies changes.**

BENCHMARK IT — NOW

In the final analysis, the thing to do is see this beautiful new system at your dealer. See its rugged professional quality. Evaluate it. Benchmark it for speed with your own routine (you'll be agreeably surprised, we guarantee you).

Find out, too, about Cromemco's reputation for quality and engineering.

Look into it now because you can have the capabilities of a fully computerized operation much quicker and for much less than you ever thought.



280 BERNARDO AVE., MOUNTAIN VIEW, CA 94040 • (415) 964-7400

Circle 87 on inquiry card.

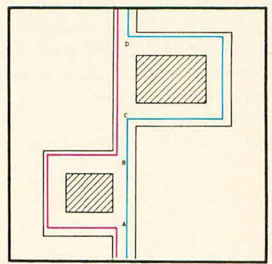
In The Queue

BYTE June 1979
Volume 4, Number 6

Foreground

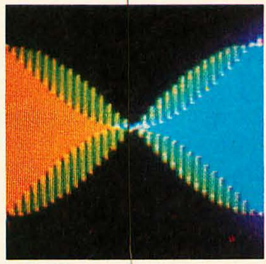
Background

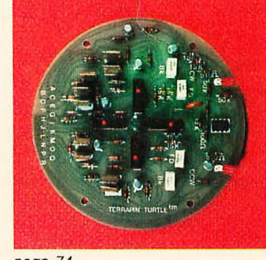
Nucleus



page 36


page 49


page 60



page 74

Cover Art:
THE TURING TEST
by Kenneth N Lodding



BYTE is published monthly by BYTE Publications Inc, 70 Main St, Peterborough NH 03458. Address all mail except subscriptions to above address: phone (603) 924-7217. Address subscriptions, change of address, USPS Form 3579, and fulfillment questions to BYTE Subscriptions, PO Box 590, Martinsville NJ 08836. Second class postage paid at Peterborough NH 03458 and at additional mailing offices—USPS Publication No. 102410 (ISSN 0360-5280). Subscriptions are $18 for one year, $32 for two years, and $46 for three years in the USA and its possessions. In Canada and Mexico, $20 for one year, $36 for two years, $52 for three years. $32 for one year air delivery to Europe. $32 surface delivery elsewhere. Air delivery to selected areas at additional rates upon request. Single copy price is $2 in the USA and its possessions, $2.40 in Canada and Mexico, $3.50 in Europe, and $4 elsewhere. Foreign subscriptions and sales should be remitted in United States funds drawn on a US bank. Printed in United States of America.

BYTE® is available in microform from University Microfilms International, 300 N Zeeb Rd, Dept PR. Ann Arbor MI 48106 USA or 18 Bedford Row, Dept PR, London WC1R 4EJ ENGLAND.

Subscription WATS Line: (800) 258-5485 Office hours: Mon-Thur 8:30 AM — 4:30 PM
Friday 8:30 AM — Noon

This month's cover by Ken Lodding is called "The Turing Test," after the famous test defined by Alan Turing. It was Turing's contention that a computer could be judged as intelligent if a human questioner could not differentiate between a computer in one room and a human being in another.

The basis for the cover painting is the Necker cube, an optical illusion where it is unclear which end of the cube is in front. The question here is: Is it the human or the computer circuit connected to the keyboard?

In This BYTE

Structured Systems business software can put a microcomputer to work for you.

SSG's general accounting, data inquiry, mailing, and communications software packages are bringing real computer power to hundreds of businesses right now. They are ready to go to work for your business.

The Honest-To-Business $12,000 Computer*

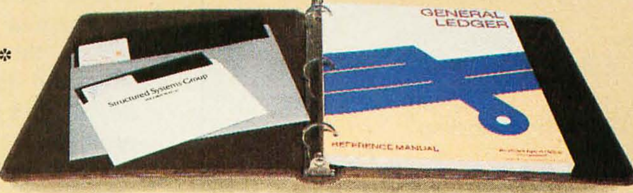
Our software will power DYNABYTE, CROMEMCO, IMSAI, NORTHSTAR, ALTOS, MICROMATION, DIGITAL SYSTEMS, or other Z-80 or 8080 based computers through your General Ledger, Accounts Receivable, and Accounts Payable. And maintain a conversational data-base query system, store and print your mailing list and labels, produce and edit correspondence, address it from your mailing list, and more. The price for a total system—hardware and SSG software—ranges from $8,000 to $14,000.

Real Business Computing

Our Business Software packages are designed to be up and running and working for you in a matter of hours. Without expensive reprogramming, technical staff additions, or costly trial-and-error. Our quality is high, our documentation practically self-instructive. The applications are flexible and extensive, designed to meet and exceed the requirements of most small to medium businesses. Real computer solutions at microcomputer prices.

Some Pleasant Surprises

Your computer retailer can give you a demonstration and literature. You might find a solution just right for your business with "off the shelf" prices and delivery times. Or we will be happy to send you literature direct, including a list of our dealers and compatible hardware. Write us, or call.



The SSG product line includes these outstanding packages:

General Ledger	LETTERIGHT Letter Writer
Accounts Receivable	NAD Mailing System
Accounts Payable	QSORT Sorting System
CBASIC-2	WHATSIT? Data/Query System

Structured Systems Group
INCORPORATED

5204 Claremont Oakland, California 94618 (415) 547-1567

* Complete prices will vary with equipment and software selected. Required: 8080 or Z-80 based computer running a CP/M or CP/M-compatible disk-based operating system. Your retailer or SSG can advise on specifics. (CP/M is a product of Digital Research.)



by Carl Helmers

On Beginning a New Project...

This week, I began a new project. It is one which could be begun by many of our readers, that of building a new computer system. In photo 1 we see what my last project turned into after four years of effort at various levels: a 6800 processor with some 28 K of programmable memory, sockets for 8 K of 2708 read only memory, a Sykes 9000 series floppy disk subsystem, a tape subsystem, and communications via parallel ports to two other computers: an Altair and my ALF Products AD-8 music synthesizer. As can be seen by the photo, this system is a packaging nightmare.

Photo 1: This homebrew personal computer system is an example of the kind of packaging nightmare which can result from experimentation with hardware. It works quite nicely, but is not exactly portable. This system proves that in hardware, as in software, it is possible to get a system where the patches and ad hoc kluges tend to outnumber the original features of the design.

I now use a cleaner machine, manufactured by Northwest Microcomputer Systems, as my primary computer. The old homebrew sits downstairs, unused for the most part. The Pascal oriented machine that is upstairs gives me a software development facility which can support my hardware projects, something I did not previously have to such a degree. However, the Pascal machine does not yet talk to the music synthesizer and the music keyboard, so I still have that problem.

To solve that problem, I have set out on a new project: building a general purpose computer for use as a local controller of the music peripherals. The communication with the main software source, the Pascal machine, will be via a high speed serial communications line when the music machine is not used alone, as in a live performance situation. The processor in this new local controller will be a Motorola 6809. It is perhaps the ultimate 8 bit processor of current technology.

In order to accomplish the musical goal of either self-contained or remote commanding of the synthesizer, such a controller must contain certain minimum functions. It must have a local communications oriented monitor, as well as a monitor oriented toward self contained operation. The communications monitor contains simple binary (not decoded ASCII) command functions for loading memory, examining memory, dumping memory, and jumping to arbitrary locations. This sort of monitor might take 100 or 200 bytes of code in the 6809's instruction set. The self-contained operations monitor

Text continued on page 124

Letters

DIGITIZING DATA BASES

Recently I saw an advertisement for the new Bit Pad One and thought of the numerous uses it presented. One that is especially interesting is building a data base.

The computer could be programmed to print a sentence or word in any form of lettering or type font, from script to Old English by letting a string equal any modified letter of the programmer's choice. With 26 strings, you would have a complete alphabet in any form you like. The computer could identify the input letter, word, or phrase, match it with the correct string variable, and print that variable (letter). The outcome would range from a letter to a full paragraph typed and printed in any font imaginable.

The only way to store such data as these modified letters without investing hours of time in plotting coordinates and typing them in, would be to illustrate the letters on the new Bit Pad One.

I hope all computer enthusiasts can derive as much enjoyment from this amazing device as I anticipate.

Jeff Korn
71 Hillary Ln
Penfield NY 14526

Any way you look at your proposed project, it is a major undertaking. The concept of building data bases from a digitizer is not new, but the programming exercise it involves is sure to be rewarding. **RGAC**

DIGITAL RADIO OPERATORS

In response to Don Stoner's letter, "Calling all Computers" (December 1978 BYTE, page 159), I thought you might be interested in some details of the new "packet radio license" available in Canada.

The Amateur Digital Radio Operator's Certificate is an Amateur Radio certificate, the holder of which is qualified to operate in some amateur radio bands. Mr. Stoner refers to this as the "Packet Radio Service" and implies that it is separate from the Amateur Radio Service. This is not true. Neither is it true that some of the band will probably go to the GRS (CB) service. As a matter of fact, the DOC seems proud of the fact that Canadian amateurs are the first in the world to implement the technique of packet radio on the amateur bands. They have made liberal bandwidth allowances in several portions of the 220 MHz band specifically for this technique, and it seems unlikely that they would start chopping off portions of this "show case" band to hand over to the GRS service.

The Amateur Digital Certificate allows operation on all amateur frequency bands above 144 MHz. This includes 144 to 148 MHz (2 meters), 220 to 225 MHz, 420 to 450 MHz, 1215 to 1300 MHz and five more bands from 2.3 to 24 GHz. It allows all current modes such as Morse code, single side band voice, FM voice, FSK or AFSK teletypewriter or data, and television, as well as several modes of pulse transmission. This is aimed primarily at the computer and electronics hobbyists who would like to participate in computer networking. The requirements (ie: examination) reflect this.

There is *no* Morse code exam at all. The written exam has three parts:

- multiple choice questions on Canadian amateur radio regulations,
- questions on radio communications theory and operation (on the Advanced Amateur level),
- the digital exam with questions on computing, analog and digital transmission, packet radio, queuing theory, digital coding, error control and other topics.

The pass requirements are 70 percent per section and the exam is not simple (I've written it), so it seems that they are looking for serious hobbyists to pass this exam.

Those who already hold an amateur or advanced amateur certificate in Canada are allowed to do anything that this new certificate allows (including packet radio) except for the pulse modes of transmission. (FSK is the current favorite for low speed networks and point to point contacts, with some type of PCM for the higher speed networks.) This new license is ideal for those computerists who want to get on the air with their terminal or computer but could never stand Morse code.

Personally, I can't wait to finish building my transmitter and get my Z-80 system on the air, and I would like to hear from other Canadian readers who are doing the same. I certainly don't talk to many hams on the HF bands who are interested in computing.

Ron Vanderhelm, VE7COR
University of British Columbia
Amateur Radio Club
Box 7 SUB, University of BC
Vancouver BC CANADA

LINEAGE PROGRAMMING

I am a genealogist and would like to get in contact with suppliers of programs for use by genealogists. I have a Digital Equipment Corp PDP-10 with expanded memory, disk, and paper tape.

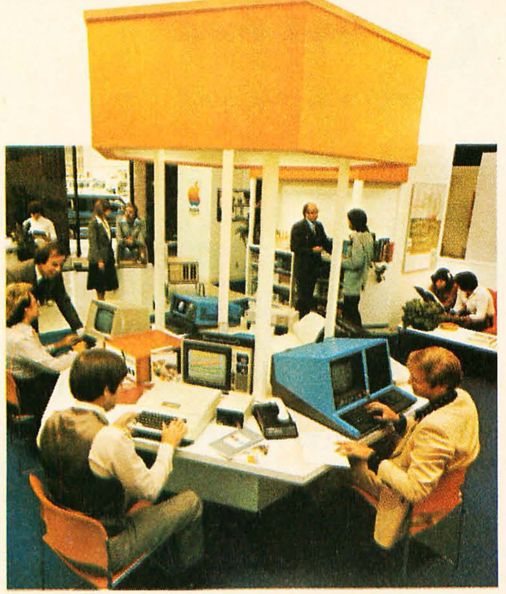
Mrs G V Creaser
4 Sunny Hill Rd
Northboro MA 01532

BEFORE YOU BUY COMPUTER #1, VISIT #1 COMPUTERLAND

If the truth is that you want a computer . . . then we want to be your computer store.

We're ComputerLand, the #1 computer store chain in the U.S. What's meaningful about that fact is, that ComputerLand has been chosen by more people as having what they've been looking for. And, since you're looking, let us tell you what you'll find, when you visit a ComputerLand store.

You'll find a product line that's continually evaluated to provide you with the widest and best selection in quality, brand name microcomputers anywhere. You'll find an enthusiastic and knowledgeable staff able to interpret all the equipment specifications, in terms of how they apply to you, and in a way you'll understand. You'll find demonstration areas where you can get a firsthand experience of running a computer yourself.



COMPUTERS FOR BUSINESS



You'll find educational materials to give you a total insight into the world of microcomputers.

You'll find a fully equipped service department to provide whatever assistance is required to keep your computer running in top-notch condition. You'll find computer user's clubs to join, where you can share ideas with people as enthusiastic as yourself. And, with each new visit, you'll find excitement—from the people you deal with, the equipment they offer, and from your own ever-growing personal involvement.

Enough about us. How about what computers do. To attempt to describe all the things your computer might do, would be to describe your imagination. So instead, we'll briefly list some of the many things for which small computers are already being used.

In business, the advent of the versatile and compact microcomputer has put the benefits of computing within reach of small companies. With systems starting at less than $6000, the businessman can computerize things like accounting, inventory control, record keeping, word processing and more. The net result is the reduction of administrative overhead and the improvement of efficiency which allows the business to be managed more effectively.

COMPUTERS FOR THE HOME



In the home, a computer can be used for personal budgeting, tracking the stock market, evaluating investment opportunities, controlling heating to conserve energy, running security alarm systems, automating the garden's watering, storing recipes, designing challenging games, tutoring the children . . . and the list goes on.

In industry, the basic applications are in engineering development, process control, and scientific and analytical work. Users of microcomputers in industry have found them to be reliable, cost-effective tools which provide computing capability to many who would otherwise have to wait for time on a big computer, or work with no computer at all.

COMPUTERS FOR INDUSTRY



And now we come to you, which leads us right back to where we started: **If you want a computer, then we want to be your computer store.**

Whether you want a computer for the home, business or industry, come to ComputerLand first. We'll make it easy for you to own your first computer. Because, simply put, we really want your business. When you come right down to it, **that's what makes us #1.**

ComputerLand®

WE KNOW SMALL COMPUTERS

ComputerLand Corp.
14400 Catalina St.
San Leandro, CA 94577
(415) 895-9363
Franchise Opportunities Worldwide.

ComputerLand Europe
Europa Terrassen
8 Rue Jean Engling
Dommeldange, Luxembourg
Phone 43 29 05 Telex 2423

A Model of the Brain for Robot Control

Part 1: Defining Notation

James Albus
Project Manager
National Bureau of Standards
United States Dept of Commerce
Washington DC 20234

The ideas presented in this article represent the views of the author and not those of the Department of Commerce or the National Bureau of Standards.

Editor's Note:

This month Dr James Albus begins an ambitious 3 part series about the brain. His theories, which evolved out of control systems theory, form an interesting contrast to Ernest W Kent's series, "The Brains of Men and Machines" in BYTE for January, February, March, and April 1978. We hope that nonmathematically oriented readers will persevere through the more technical sections in order to benefit from Dr Albus's insights. . . . CM

In order to build a computer model of the brain for robot control we must start with a clear understanding of what the brain is for (ie: its primary function). If one examines what most brains do all of the time, and what our own brains do most of the time, it is clear that the brain is *not* used primarily for thinking.

The brain is first and foremost a control system. All brains, even that of the tiniest insect, control behavior. Some brains can produce very complex behavior, but only the most sophisticated and highly developed brains exhibit the phenomenon of thought. Clearly then, thought is not the central purpose of the brain, but is, rather, an artifact that arises out of the complex computing mechanisms required to generate and control extremely sophisticated behavior.

This implies that would-be brain modelers should first attempt to understand, and if possible, reproduce the control functions and behavior patterns that exist in insects, birds, mammals, and, in particular, primates. Only after these control systems are successfully modeled can we expect to understand the mechanisms that give rise to intelligence and abstract thought in the human brain.

If the brain is primarily a control system, then any brain model we construct should control something. One of the most obvious candidates is a robot manipulator, since it rather closely resembles a limb, the most common type of device controlled by the brain. We shall therefore first develop a computer model of a basic neurological structure which can compute control functions for a robot manipulator.

We shall then attempt to demonstrate how this basic model can be generalized to compute a broad class of analytic, transcendental, or logical functions and production rules of many multivalued variables. We will show how this same model can learn, remember, and recognize patterns and how it can be interconnected into a hierarchical network for generating sensory interactive, goal directed behavior.

We will suggest how such a hierarchy might remember experiences, solve problems, plan tasks, select goals, answer questions, structure knowledge of the world and events, and understand and generate music or natural language. Finally, we will also suggest some possible experiments and lines of research that might be pursued by one or more ambitious personal computer enthusiasts with limited resources.

The Nature of Computation in the Brain

The brain is, of course, not a single computer, but rather a network of billions of individual computing devices interconnected so as to produce coordinated and unified action. There are millions of photodetectors in each eye and thousands of audio detectors in each ear. The body is embedded with sensors which detect touch, pressure, heat, cold, and pain; chemical analyzers that detect the smell and taste of things; and sensors that measure the position of joints, the tension in tendons, and the length and velocity of contraction of muscles. Inertial sensors measure roll, pitch, and yaw accelerations, and the position of the head with respect to gravitational attraction; and hormone detectors, thermosensors, and blood chemistry analyzers report on the internal biological condition of the organism.

All of this information is analyzed and processed in innumerable computing centers which detect patterns, compare incoming data with stored expectations, and evaluate

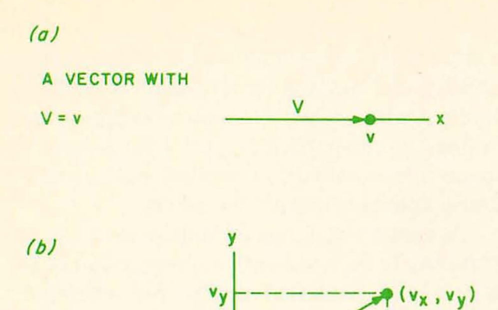
the results. In many different ways and at many different levels this sensory data stream interacts with the behavior generating system to select goals, modify habits, and direct the actions of millions of muscles and glands to produce what is observed as behavior.

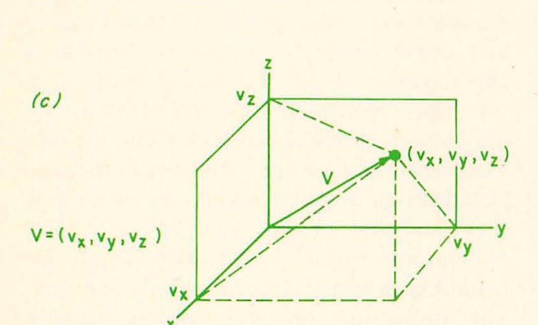
Perhaps the most obvious feature of the brain is that many computations are going on in many different places simultaneously. The brain does not execute sequential programs of instructions under control of a program counter. There is no fetch/execute cycle. The mathematics of finite state automata and Turing machines are not well-suited for describing the basic operations of the brain. In fact, the fundamental computations performed in the brain are not even digital — they are analog. Each neuron in the brain is essentially an analog computer performing complex additions, integrations, differentiations, and all sorts of nonlinear operations on input variables that can number from one to several hundred thousand.

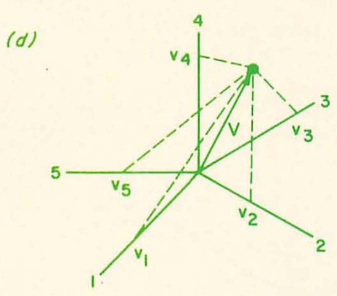
The brain is a digital device only in that information is encoded for transmission from one neuron to another over long transmission lines (called *axons*) by pulse-frequency or pulse-phase modulation. When these pulse encoded signals reach their destinations, they are reconverted into analog voltages for the computations which take place in the dendrites and cell bodies of the receiving neurons (see "Designing a Robot from Nature" February 1979 BYTE, page 28).

The brain achieves its incredible precision and reliability through redundancy and statistical techniques. Many axons carry information concerning the value of the same variable, each encoded slightly differently. The statistical summation of these many imprecise and noisy information channels results in the reliable transmission of precise messages over long distances. In a similar way, a multiplicity of neurons may compute on roughly the same input variables. Clusters of such computing devices provide statistical precision and reliability orders of magnitude greater than that achievable by any single neuron. The outputs of such clusters of neurons are transmitted and become inputs to other clusters, which perform additional analog computations. These are the variables we have to deal with and the computations we have to simulate if we are to model the brain in any meaningful way.

To those familiar only with fetch/execute machines, this may seem an extremely difficult structure to model. I hope, in the course of these articles, that some of the difficulties







(c)

V = (vx, vy, vz)

Figure 1: Defining space with vectors. A vector is an ordered list of variables which defines a point in space; (a), (b), (c), and (d) depict vectors representing 1, 2, 3, and 4 dimensions, respectively. The number of dimensions in the space is equal to the number of variables in the list. (The illustration in (d) is meant only to be symbolic of a four-dimensional vector, which cannot be visualized in three dimensions.)

will be cleared away and the prospects for building such structures will seem less dubious.

The Need for Notation

In order to discuss an engineering design for a robot control system modeled after the brain, we must first devise a mathematical convention and notation to bridge the gap between the structure of the brain and the structure of currently available computers. This is essential if we are to describe behavior precisely and to translate that description into a design for circuits and program statements to generate behavior in a computationally concise manner.

Vectors

One way to describe many variables and deal with many simultaneous multivariant computations is to use *vector* notation. A vector is simply an ordered set, or list of variables. A vector can specify magnitude and direction. The vector **V** in figure 1b has two components v_x along the X axis and v_y along the Y axis. The ordered set, or list of components define the vector so that we can write $V = (v_x, v_y)$.

The components of a vector can also be considered as the coordinates of a point (v_x, v_y) which corresponds to the tip of the vector. The locus of all pairs of components which can exist defines a vector space (for two dimensions the vector space is a surface). A vector can have more than two

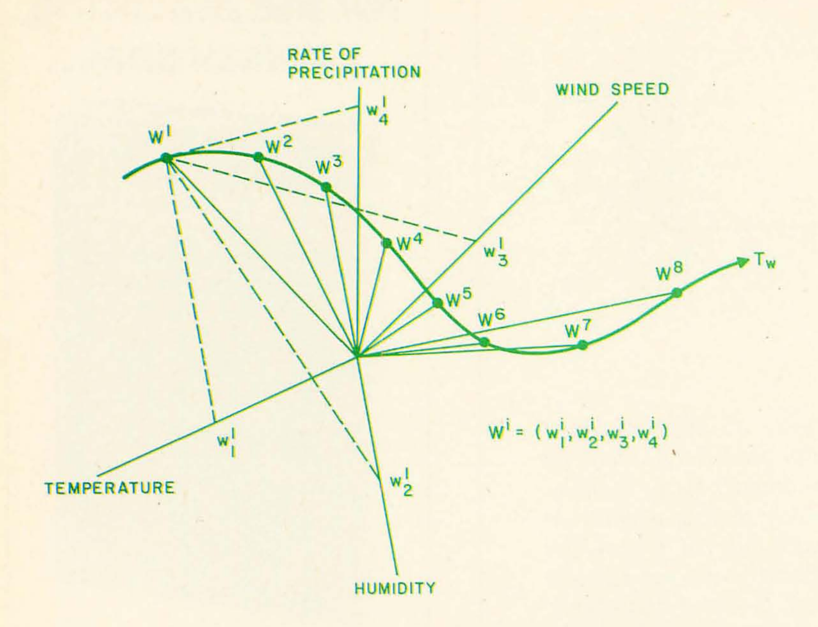


Figure 2: If, as time progresses, one or more of the components of a vector W *change, the vector will move through space, tracing out a trajectory* T_W.

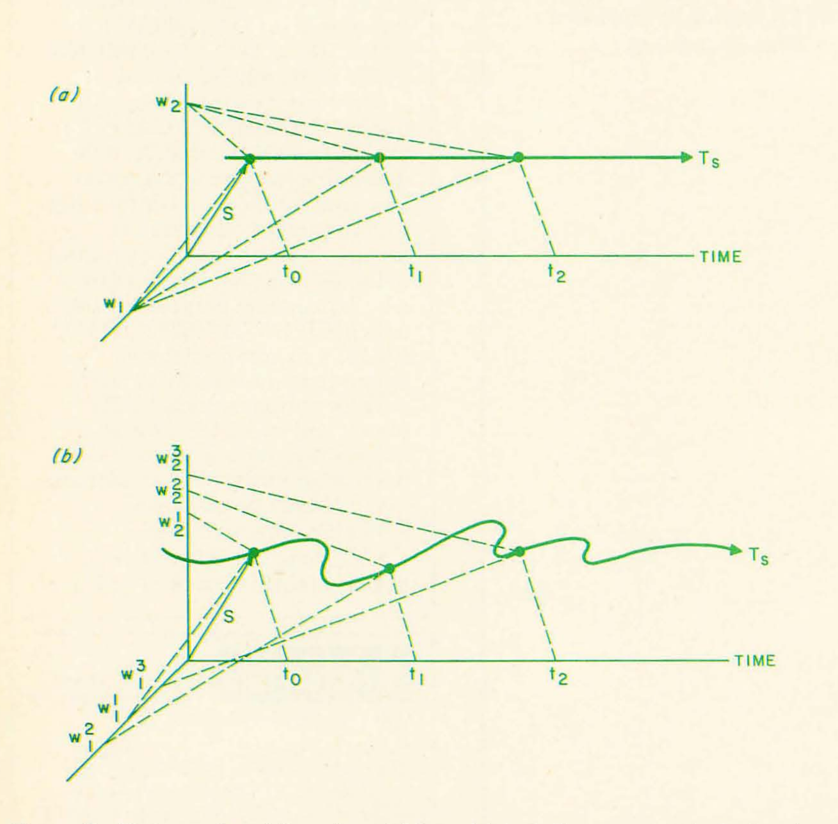


Figure 3: If the ordered list of variables which define a vector includes time, the space defined by the vector will have time as one of its axes. As time progresses the vector will move along the time axis. If none of the other variables is time dependent, the trajectory will be a straight line parallel to the time axis, as in (a). If any of the other variables change with time, the trajectory will be some curve with a component along the time axis as in (b).

components. A vector with three components defines a volume (figure 1c), and a vector with four or more components defines a hyperspace (figure 1d). A hyperspace is impossible to visualize, but is a very useful concept for our discussion.

A vector in a higher dimensional space can usually be visualized as a projection onto a lower dimensional space. For example, typical mechancial drawings portray front, side, and top views of a three-dimensional form projected onto a two-dimensional sheet of paper. Each projection can either illustrate a cut through the object at a particular plane along the projection axis, or a superposition of all the salient features of the object collapsed into the plane of the illustration. In the collapsed version, the fact that two points or lines intersect in the projected image does not necessarily mean that they coincide or intersect in the higher dimensional space — they may simply lie behind each other along the projection axis. The projection operator ignores variable differences which correspond to distance along the projection axis.

It is not necessary to make the projection axis coincident with any of the coordinate axes. For example, in the oblique projection (perspective drawing) of figure 1c, the projection axis (the normal line to the paper through the origin of the coordinate system) is not aligned with any of the coordinate axes. The lines in the drawing represent the projections of lines in a three-dimensional space onto the two-dimensional surface of the paper. In a similar way we can project higher dimensional vectors and hyperspaces of any dimension onto a two-dimensional drawing. Figure 1d illustrates a four-dimensional vector projected onto a two-dimensional drawing.

States and Trajectories

A vector can specify a state. This is the primary use we shall make of vectors in this discussion. A state is defined by an ordered set of variables. For example, the state of the weather might be characterized by a state vector $W = (w_1, w_2, w_3, w_4)$ where:

w_1 = temperature,
w_2 = humidity,
w_3 = wind speed,
w_4 = rate of precipitation.

Now the weather, like many things, is not constant. It varies with time. Each of the state variables (temperature, humidity, wind speed, and rate of precipitation) is time dependent. Thus, as time passes, the point defined by W^t will move through the four-

dimensional space. Figure 2 illustrates the locus of the point traced out by **W** as it moves to define a trajectory $\mathbf{T}_w$.

It will often be convenient to represent time explicitly in our notation. We can easily do this by simply adding one more variable, time (t), to our state vector, thus increasing by one the number of dimensions in the space defined by the state vector. For example $W = (w_1, w_2, w_3, w_4, t)$.

As time progresses, any point defined by the state vector moves along the time axis. A state vector whose w_i components do not vary with time will now trace out a straight line trajectory, parallel to the time axis as shown in figure 3a. If, however, any of the w_i components is time dependent, the state trajectory will contain velocity components that are orthogonal, as well as parallel to the time axis, as shown in figure 3b.

If we project the state space of all the variables except time onto a two-dimensional surface, we can represent the passage of time by the motion of this two-dimensional plane along the time axis normal to it, as in figure 4. The state trajectory $\mathbf{T}_s$ is the locus of points traced out by the state vector as time passes.

A large variety of things can be represented as vectors. For example, we can represent an ASCII character as a vector (figure 5). The ordered set of binary digits in the ASCII representation corresponds to the components of a binary vector. Each symbol in the ASCII alphabet is uniquely paired with a vector in an eight-dimensional hyperspace. Each symbol thus corresponds to a point in the hyperspace.

This is an important concept, because it allows us to define any set of symbols as vectors or points in hyperspace. Any string of symbols then becomes a trajectory through the hyperspace. For example, the string of symbols, "the cat chased the rat," can be described as a trajectory through a hyperspace defined by any set of variables defining the English alphabet (plus a blank character). This also applies to the string WXYZ when:

W is the command: Reach to Position A;
X is the command: Grasp;
Y is the command: Move to Position C;
Z is the command: Release.

We need not restrict ourselves to binary vectors. Symbols may be represented by vectors with continuously variable components as well. This allows us to introduce the concept of *fuzzy* symbols. If the hyperspace is continuous, then each point which corresponds to a symbol has some neighbor-

About the Author:

Dr James S Albus worked for NASA from 1957 to 1972 designing optical and electronic subsystems for over 15 spacecraft, and for one year managed the NASA Artificial Intelligence Program. Since 1973 he has been with the National Bureau of Standards where he has received several awards for his work in advanced computer control systems for industrial robots. He has written a survey article on robot systems for Scientific American *(February 1976) and his Cerebellar Model Arithmetic Computer won the* Industrial Research Magazine *IR-100 Award as one of the 100 most significant new products of 1975.*

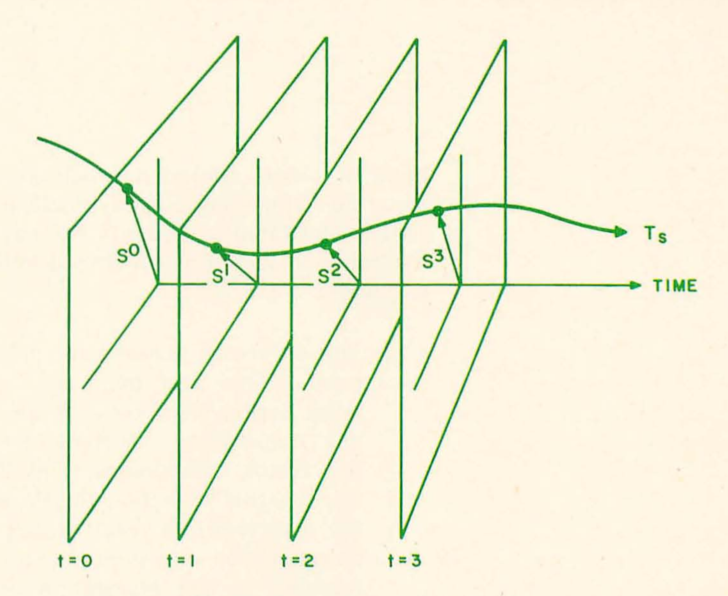


Figure 4: If the vector space defined by all of the vector components except time is projected upon a two-dimensional surface, then the passage of time can be represented as the movement of the two-dimensional surface along the time axis normal to it.

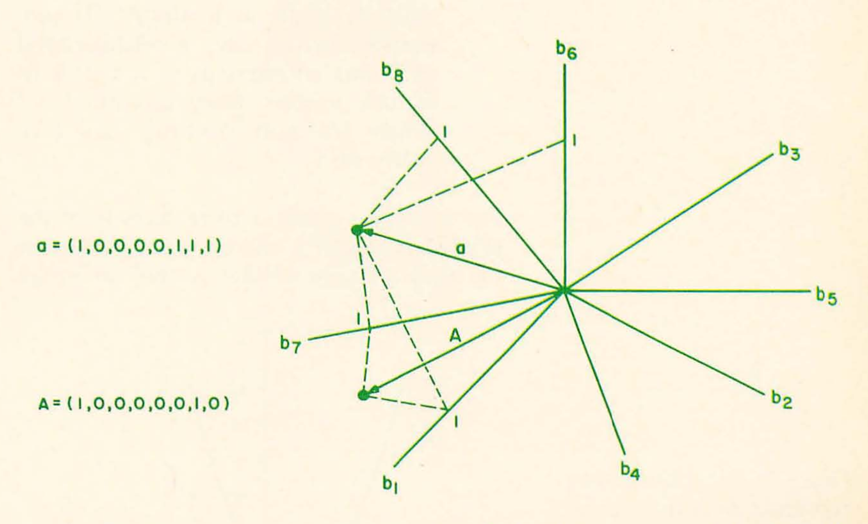
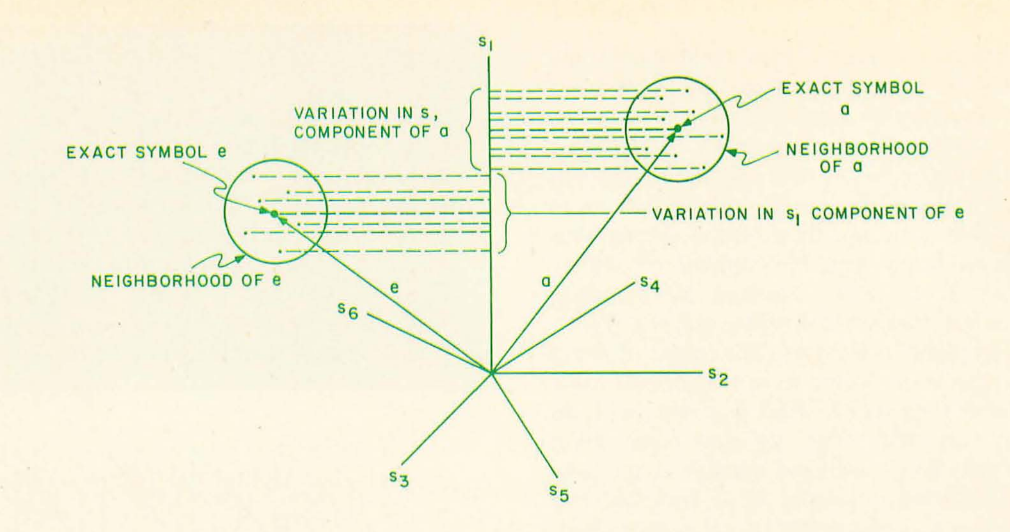


Figure 5: A vector can represent a symbol. Here two symbols from the ASCII character set, an uppercase A and a lowercase a, are represented as vectors (or points) in an eight dimensional space. The values of the eight bits in the ASCII code are plotted along the eight axes. (b_8 is the even parity bit.)

Figure 6: Each point in hyperspace, corresponding to a particular symbol such as a or e, has some neighborhood of points around it which are closer to it than to any other symbol. Variations from the exact, or ideal position of a symbol vector may derive from noise in a transmission channel or from differences between the observed symbol and the ideal.



hood of points around it which are much closer to it than any other symbol's points. This is illustrated in figure 6. We may view the points in such a neighborhood in one of two ways:

1. The difference between the neighborhood points and the exact symbol point derives from noise on the channel transmitting variables denoting the vector components. This is useful in signal detection theory, where the detection of a vector within some neighborhood of a symbol vector corresponds to the recognition of that symbol against a noisy background.
2. The difference from the exact symbol derives from distortions or variations in the symbol itself. This makes the best sense if the components of the symbol's vector are *values* of attributes or features of the symbol, rather than arbitrary digits as in the ASCII convention. In this case, a neighborhood of points corresponds to a cluster of feature vectors from a symbol set which are not identical, but very nearly so.

For example, a vector of features from the printed character *e* will be slightly different for each instance of that symbol on a page due to variations in the paper on which it is printed. However, if these *e* feature vectors fall in compact clusters far from the feature vectors of other symbols, the letter *e* will be easily recognized, despite the fact that no two specimens are exactly alike.

This is a fundamental concept in pattern recognition theory. Hyperspace is partitioned into regions, and the existence of a feature vector in a particular region corresponds to the recognition of a pattern or symbol. By definition, the best set of features is the one that maximizes the separability of pattern vectors. In the design of pattern recognizers it is important to select a set of features which is easily measured and which produces widely separated and compact clusters in feature space.

Functions and Operators

In the physical world, functions are usually defined as relationships between physical variables. For example, we could say that climate over a particular geographical region is a function of the heat input, the prevailing wind conditions, and other factors, or that the seasons are a function of the position and orientation of the earth relative to the sun. Similarly, we may say that the level of hunger we experience is a function of the signals on nerve fibers reporting on the state of the stomach, chemistry of the blood, the time of day as indicated by internal biological rhythms, and so on.

In mathematics a function defines (and is defined by) a relationship between symbols that can sometimes be set in one-to-one correspondence to physical variables. As in the physical world, a function usually implies a directional relationship (eg: the relationship between cause and effect has a direction which flows from cause to effect). In traditional terms a function may be expressed as an equation, such as:

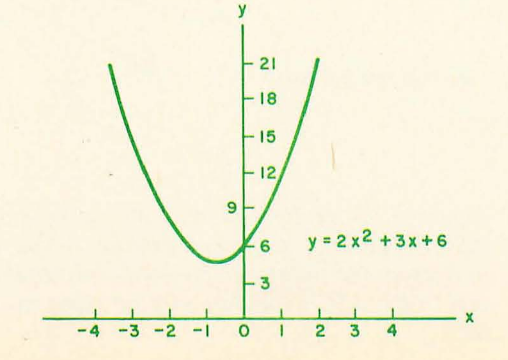


Figure 7: Functions can be expressed in a number of different ways. Here the functional relationship between Y and X is expressed as an equation and a graph.

$$y = f(x)$$

which reads: y equals a function f of x. The function:

$$y = 2x^2 + 3x + 6$$

is a relationship between y and x.

Functions can also be expressed as graphs. Figure 7 is a plot of the equation $y = 2x^2 + 3x + 6$. Functions may sometimes be defined by tables. The table in figure 8a defines the Boolean AND function $Z = X \cdot Y$. This function can also be drawn as a circuit element (see figure 8b) which performs the AND function on two inputs.

Tables can also be used to define non-Boolean functions. Tables of logarithms or trigonometric functions are good examples of this. Of course, a table defines a continuous function exactly *only* at the discrete points represented in the table. Thus, the accuracy of a continuous function represented by a table depends on the number of table entries (ie: the resolution on the input variables). Accuracy can, of course, be increased by interpolation techniques. In general, the number of entries required to compute a function by a table lookup is proportional to R^N, where R is the resolution of each input variable, and N is the number of input variables. This exponential increase in size of the table required is the principal reason that multidimensional functions are seldom computed by table lookup.

Modern mathematics often expresses functional relationships in terms of *mappings* from a set of states defined by independent variables onto a set of states defined by dependent variables. In one notation, this is expressed by the string f:

$$f: C \rightarrow E$$

which reads, "f is a relationship which maps the set of causes C into the set of effects E." It means that for any particular state in the set C, the relationship f will compute a state in the set E. This is shown in figure 9.

We have already shown that states can be denoted by vectors and sets of states by sets of points in vector hyperspaces. Thus, the notion of a function being a mapping from one set of states to another naturally extends to a mapping of points in one vector hyperspace onto points in another.

Suppose, for example, we define an operator h as a function which maps the input $S = (s_1, s_2, s_3, \ldots s_N)$ onto the output scalar variable *p*. We can write this as:

$$p = h\,(S)$$

or

$$p = h\,(s_1, s_2, \ldots s_N)$$

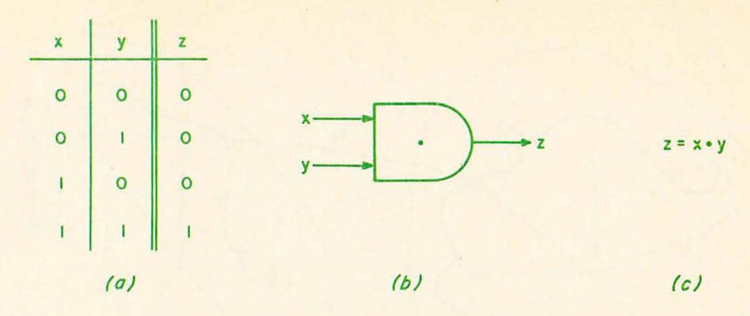


Figure 8: Functions can also be expressed as tables and circuits. Here the Boolean function $Z = X \cdot Y$ is expressed as a table, a circuit, and an equation.

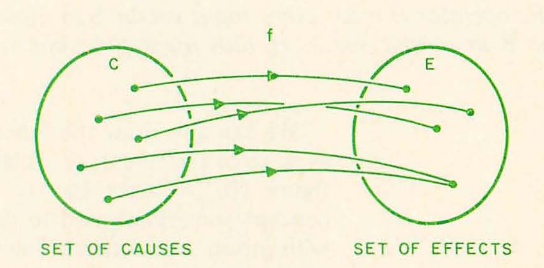
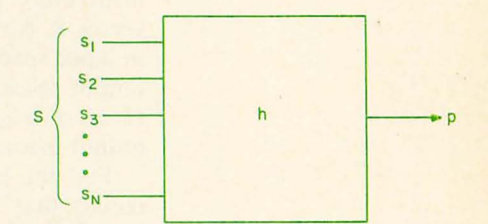


Figure 9: A function can also be expressed as a mapping from one set onto another. Here the function **F** *maps the set of causes* **C** *onto the set of effects* **E** *such that for every cause in* **C** *there is an effect in* **E**. *In our discussion we will be concerned only with single valued functions such that there is only one effect for each cause. We will, however, allow more than one cause to have the same effect (ie: more than one point in* **C** *can map onto the same point in* **E**).

Figure 10: We will define the operator h as a function which maps the input vector **S** *into the output scalar variable p.*



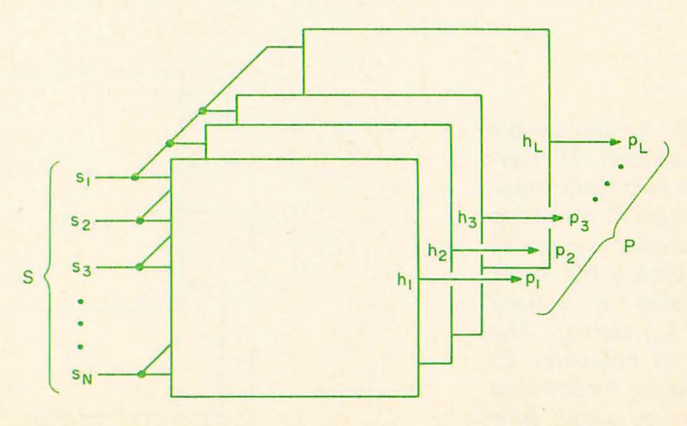


Figure 11: We will define the set of operators $H = (h_1, h_2, \ldots h_L)$ as a function which maps the input vector **S** *into the output vector* **P**.

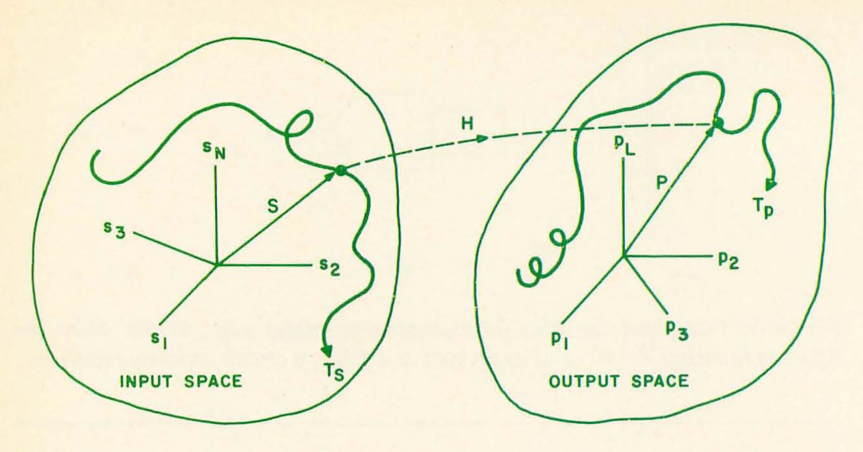


Figure 12: The operator H maps every input vector **S** *in input space into an output vector* **P** *in output space. H thus maps the trajectory* $\mathbf{T}_s$ *into the trajectory* $\mathbf{T}_p$*.*

We can also draw the functional operator as a circuit element or "black box" as in figure 10. (A black box is an engineering concept sometimes used to depict a process with inputs and outputs. The viewer sees the effects on the output of changes to the input, but the internal workings of the process remain hidden in a black box.)

If we assume that we have L such operators, $h_1, h_2, \ldots h_L$, each operating on the input vector **S** in figure 11, we have a mapping:

$$H: \mathbf{S} \rightarrow \mathbf{P} \text{ or } \mathbf{P} = H(\mathbf{S})$$

where the operator $H = (h_1, h_2, \ldots h_L)$ maps every input vector **S** into an output vector **P**. Now since **S** is a vector (or point) in input space and **P** is a vector (or point) in output space, we can represent the function H as a mapping from input space onto output space, as shown in figure 12.

For the purposes of our discussion we require that both the input and output space be bounded and that each **S** will map into one and only one **P**. Several different **S** vectors may map into the same **P** vector, however. Of course, if any of the variables in **S** are time dependent, **S** will trace out a trajectory $\mathbf{T}_s$ through input space. The operator H will map each point **S** on $\mathbf{T}_s$ into a point **P** on a trajectory $\mathbf{T}_p$ in output space.

Goal Seeking Control Systems

We are now ready to consider the structure of control systems for sensory interactive, goal directed behavior. The simplest form of goal seeking device is the *servomechanism.* The setpoint, or reference input to the servomechanism, is a simple form of command. Feedback from a sensing device, which monitors the state of the output or the results of action produced by the input, is compared with the command. If there is any discrepancy between commanded action and the results, an error signal is generated which acts on the output in the proper direction and by the proper amount to reduce the error. The system thus follows the setpoint, or, put another way, it seeks the goal set by the input command.

Now almost all servomechanism theory deals with a one-dimensional command, a one-dimensional feedback, and a one-dimensional output. Our vector notation will allow us to generalize from this one-dimensional case to the multidimensional case with little difficulty.

Assume we have the multivariable servomechanism shown in figure 13. The function H operates on the input variables in **S** and computes an output **P** = H(**S**). Note that we have partitioned the input vector **S** into two vectors: $\mathbf{C} = (s_1, s_2 \ldots, s_i, 0 \ldots, 0)$ and $\mathbf{F} = (0, \ldots 0, s_{i+1}, \ldots s_N)$; such that **S** = **C**+**F**. If i = 1, N = 2, L = 1, and H computes some function of the difference between **C** and **F**, we have a classical servomechanism.

In our more general case **C** may be any vector, and in some cases it may be a sym-

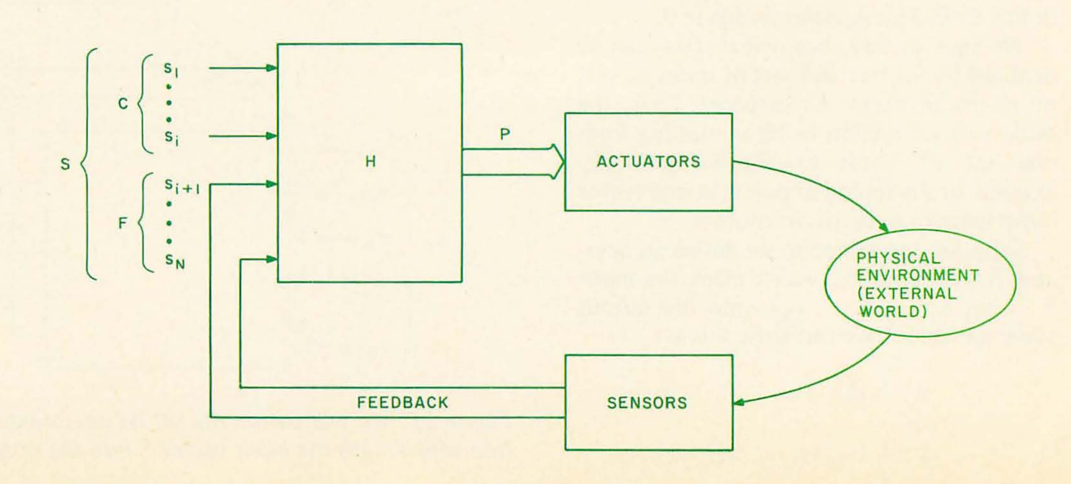
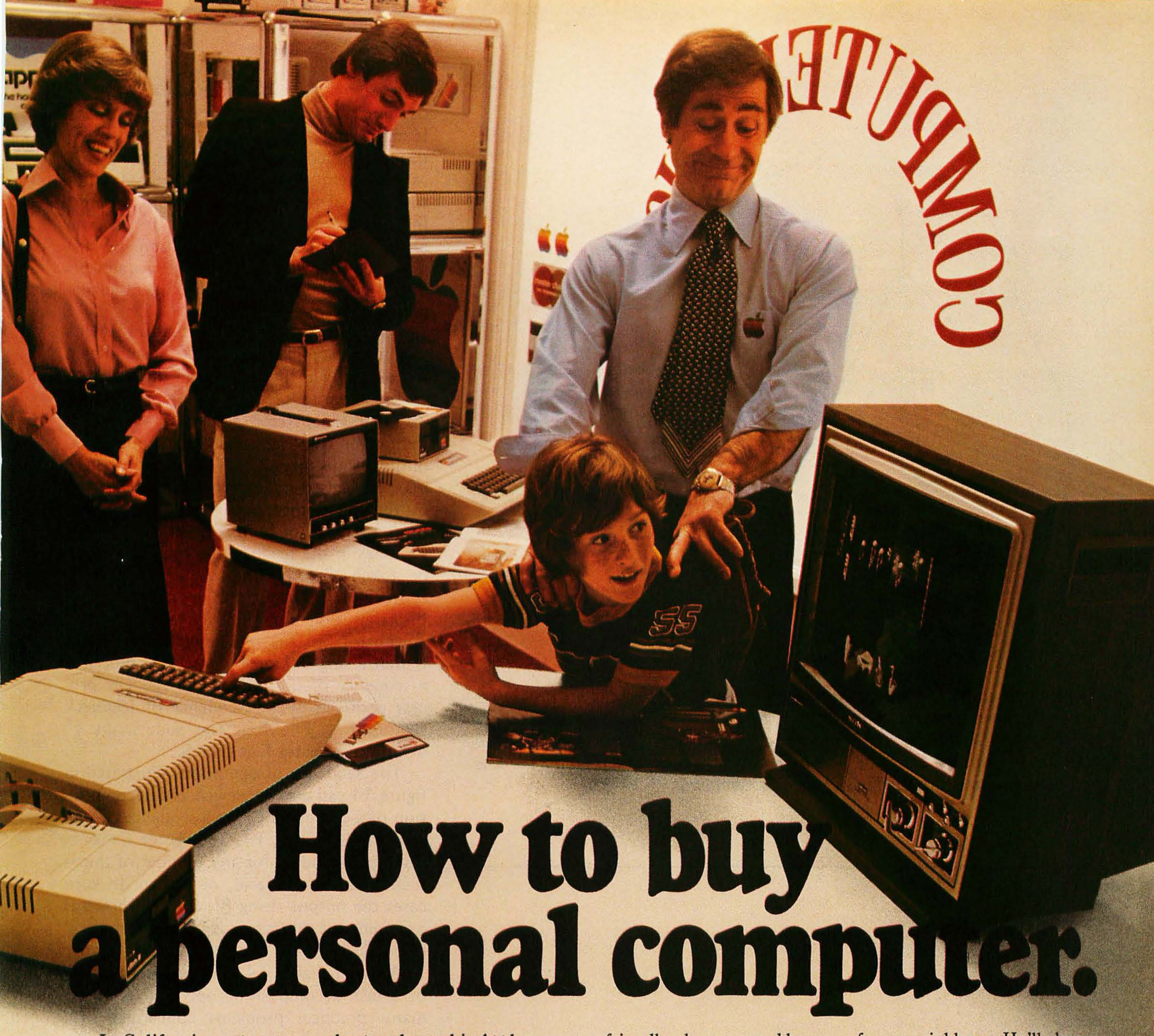


Figure 13: A multivariable servomechanism. The reference, or command input is the vector **C** *consisting of the variables* s_1 *thru* s_i*. The feedback is the vector* **F** *consisting of sensory variables* s_{i+1} *thru* s_N*. The function H computes an output vector* **P** *consisting of* p_1 *thru* p_L *which drive actuators and thus affect the physical environment.*



How to buy a personal computer.

In California, a store owner charts sales on his Apple Computer. On weekends though, he totes Apple home to help plan family finances with his wife. And for the kids to explore the new world of personal computers.

A hobbyist in Michigan starts a local Apple Computer Club, to challenge other members to computer games of skill and to trade programs.

Innovative folks everywhere have discovered that the era of the personal computer has already begun—with Apple.

Educators and students use Apple in the classroom. Businessmen trust Apple with the books. Parents are making Apple the newest family pastime. And kids of all ages are finding how much fun computers can be, and have no time for TV once they've discovered Apple.

Visit your local computer store

The excitement starts in your local computer store. It's a friendly place, owned by one of your neighbors. He'll show you exactly what you can use a personal computer for.

What to look for

Your local computer store has several different brands to show you. So the salesman can recommend the one that best meets your needs. Chances are, it will be an Apple Computer. Apple is the one you can program yourself. So there's no limit to the things you can do. Most important, Apple's the one with more expansion capability. That means a lot. Because the more you use your Apple, the more uses you'll discover. So your best bet is a personal computer that can grow with you as your skill and involvement grow. Apple's the one.

It's your move

Grab a piece of the future for yourself. Visit your local computer store. We'll give you the address of the Apple dealer nearest you when you call our toll-free number. Then drop by and sink your teeth into an Apple. **800-538-9696**. In California, **800-662-9238**.

apple® computer

Circle 15 on inquiry card.

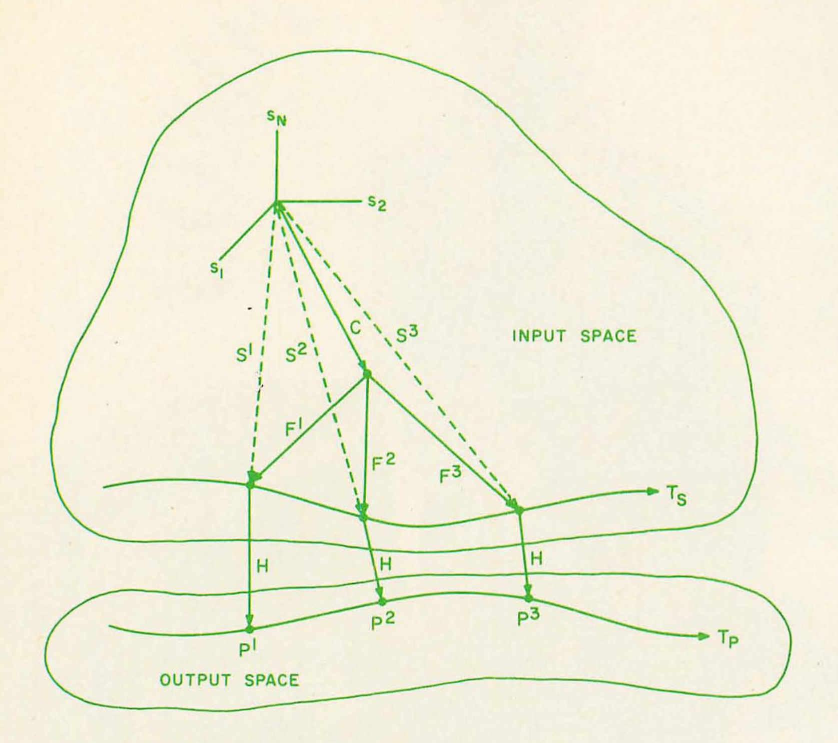


Figure 14: A stationary **C** *vector establishes a setpoint, and as time progresses the feedback vector varies from* F^1 *to* F^2 *to* F^3. *The* **S** *vector thus traces out a trajectory* T_s. *The H operator computes an output* **P** *for each input* **S** *and so produces an output trajectory* T_p. *The result is that the input command* **C** *is decomposed into a sequence of output subcommands* P^1, P^2, P^3.

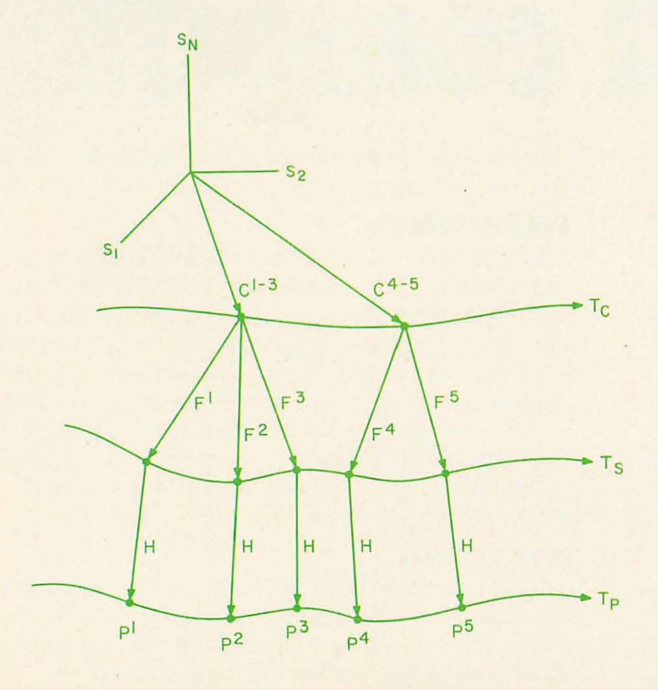


Figure 15: If the command vector **C** *also changes from time to time, it will trace out a trajectory* T_C.

bolic command. The feedback vector may contain information of many different types. It may simply report position or velocity of the controlled outputs, but for a complicated system such as a robot manipulator or the limb of an animal, it may also report the resistance to movement by the environment, the inertial configuration of the manipulator structure, and other parameters relevant to the problem of making rapid and precise movements.

Figure 14 illustrates the situation when a stationary command vector **C** establishes a setpoint, and as time progresses the feedback vector **F** varies, creating an input trajectory T_s. The H operator computes an output vector for each input and so produces an output trajectory T_p. The variation in **F** may be caused by external forces imposed by the environment, or by actions produced by the output, or both. One or more of the variables in the feedback vector **F** may even be taken directly from the output vector **P**. In the latter case the H operator becomes the transition function for a finite state automaton. In any of these cases the result is that a single command vector **C** produces a sequence of output vectors T_p. The process is driven by the sequence of feedback vectors F^1, F^2, F^3. The superscript F^k denotes the vector **F** at time t_k.

The sequence of operations illustrated in figure 14 can also be viewed as a decomposition of a command **C** into a sequence of subcommands P^1, P^2, P^3. The vector **C** may be a symbol standing for any number of things such as a task, a goal, or a plan. In such cases the output string P^1, P^2, P^3 represents a sequence of subtasks, subgoals, or subplans, respectively.

Whether figure 14 is a servomechanism or a task decomposition operator, there are many practical problems concerned with stability, speed, gain, delay, phase shift, etc. In our notation these are all embedded in the H functions. If the H functions are correctly formulated and defined over the entire space traversed by the **S** input, then the output T_P will drive the physical actuators in such a way that the goal is achieved (ie: the error between the command **C** and the result **P** is nulled) and stability is maintained under all conditions.

Servomechanisms are, of course, only the simplest form of sensory interactive, goal seeking devices. By themselves they are certainly not capable of explaining the much more complex forms of goal seeking commonly associated with purposive behavior in biological systems. However, when connected together in a nested (or hierarchical) structure, the complexity of behavior in feedback control systems increases dramatically.

Hierarchical Control

Assume that the command vector C in figure 14 changes such that it steps along the trajectory T_c as shown in figure 15. The result is that the sequence of input commands C^1, C^2, C^3, followed by the sequence C^4, C^5 produces the sequence of output vectors P^1, P^2, P^3, P^4, P^5. In this case the subsequence P^1, P^2, P^3, is called by the commands C^1, C^2, C^3 and driven by the feedback F^1, F^2, F^3. The subsequence P^4, P^5 is called by C^4, C^5 and driven by F^4, F^5, etc.

If we now represent time explicitly, the C, F, and P vectors and trajectories of figure 15 appear as shown in figure 16. The fact that C remains constant while the feedback changes from F^1 to F^2 to F^3 means that the trajectory T_c is parallel to the time axis over that interval. The jump from C^1, C^2, C^3 to C^4, C^5 causes an abrupt shift in the T_c trajectory in the time interval between F^3 and F^4.

Note that each instant can be represented by a plane (or set of coplanar regions) perpendicular to the time axis. Each plane contains a point from each trajectory and represents a snapshot of all the vectors simultaneously at a specific instant in time.

We are now ready to consider a hierarchy of servomechanisms, or task decomposition operators, as shown in figure 17a. Here the highest level input command C_4 is a symbolic vector denoting the complex task (ASSEMBLE AB). Some of the components in C_4 may denote modifiers and arguments for the assemble task. The subscript C_k denotes the C vector at the k^{th} level in the hierarchy.

Note that in figure 17 vectors are not repeatedly drawn for each instant of time during the trajectory segments, when they are reasonably constant. Thus, C_4 is shown only at the beginning and end of the trajectory segment labeled (ASSEMBLE AB). C_2 is shown only at the transition points between (REACH to A), (GRASP), (MOVE TO C), etc. It should be kept in mind, however, that H_4 computes P_4 continuously and produces an output at every instant of time, just as H_1 computes P_1.

The feedback F_4 may contain highly processed visual scene analysis data which identifies the general layout of the work space, and thereby determines which output vectors P_4 (and hence which simple task commands C_3) should be selected and in which order. F_4 may also contain data from P_4 and P_3 which indicates the state of completion of the decomposition of C_4. F_4 combines with C_4 to define the complete input vector S_4. The H_4 operator produces an output vector $P_4 = H_4 (S_4)$.

Text continued on page 24

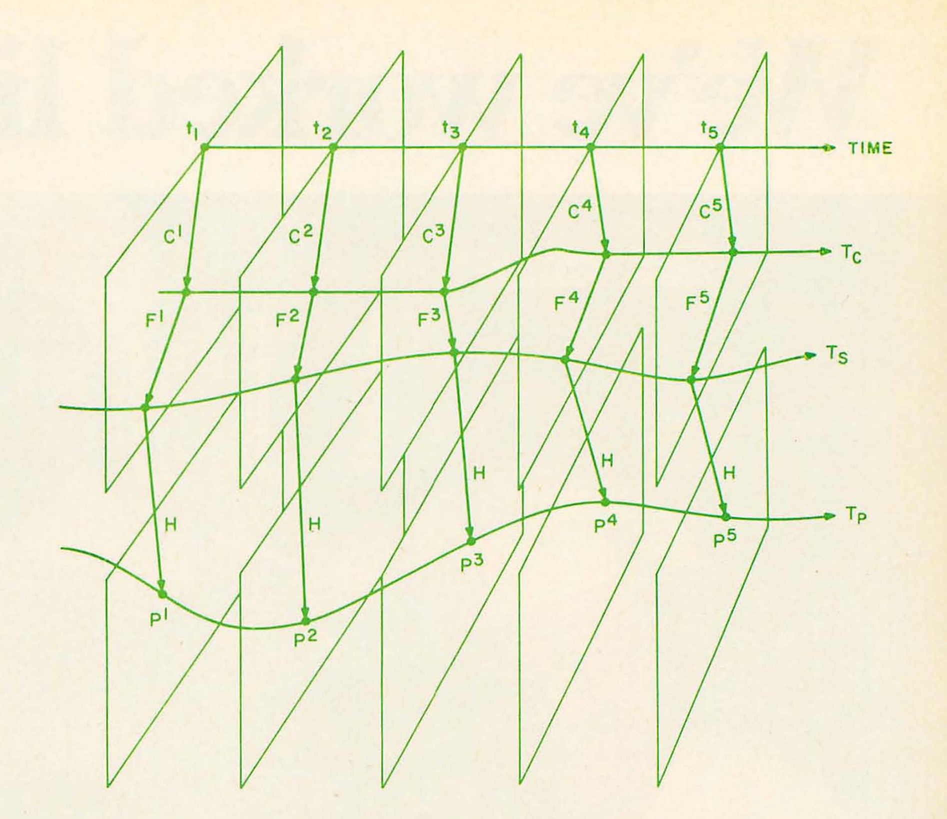
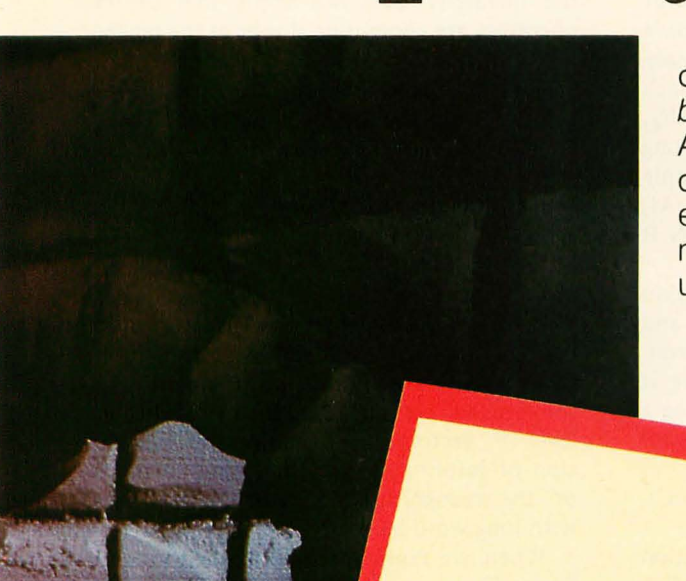


Figure 16: When time is represented explicitly, the vectors and trajectories of figure 15 become as shown here. In this example, the C *vector remains constant from time t = 1 to t = 3 and then jumps to a new value for t = 4 and t = 5.*

We've worked like mad to



cook up all your favorites.



Now, we're cooking. Our boys in the lab have turned circuit chefs these past three months to create a *smorgasbord* of deliciously assembled boards to support your APPLE II,* TRS-80‡ or S-100 bus systems. Feast your eyes on our monstrous selection in the menu below. Then, order enough to satisfy your hunger for experimentation for months to come. Contact your local computer store, or call us direct. *Bon appetit!*

California Computer Systems
309 Laurelwood Road · Santa Clara, CA
(408) 988-1620 · 95050

So Nobody Goes Away Mad.

MENU

APPLE II Delicacies

7114 APPLE ROM **$69.95**
14K ROM . . . uses 7-2716 + 5V only . . . selectable ROM overlay . . . powers down . . . offers BLOCK and INDIVIDUAL ENABLE . . . permits DMA & INTERRUPT daisy chain operation.

7740 APPLE Programmable Timer Module **$150.00**
An MC6840 programmable computer timer for use in frequency measurements . . . event counting . . . interval measuring . . . square wave generation . . . gated delay signalling . . . creating single pulses of controlled duration . . . pulse width modulation . . . and system interrupts.

7490 APPLE GPIB IEEE 488 Interface **$250.00**
Interface offering selectable unit addressing . . . supports bus as a talker, listener and controller . . . includes on-board ROM.

7710 APPLE Asynchronous Serial Interface **$99.95**
Interface featuring: on-board ROM . . . accurate baud rates . . . all common baud rates from 50 baud through 19200 baud . . . external baud available . . . conforms to EIA RS404 and a subset of RS232C.

7712 APPLE Synchronous Serial Interface **$99.95**
Interface featuring: on-board ROM . . . accurate baud rates . . . all common baud rates from 50 baud through 19200 baud . . . external baud available . . . conforms to EIA RS404 and a subset of RS232C.

7720 APPLE Parallel Interface **$69.95**
Interface with dual PIA . . . on-board ROM . . . power down ROM . . . supports INTERRUPT daisy chain.

7811 APPLE Arithmetic Processor . . **$399.00**
Processor with on-board, power-down ROM . . . 32-bit floating point arithmetic operation . . . based on AMD 9511 device.

7470 APPLE 3¾ Digit BCD Analog-to-Digital Converter **$129.00**
±0 to 4.999 VDC range . . . floating or ground reference.

7520 APPLE Extender **$25.00**
7500 APPLE Wire Wrap Board **$21.00**
7590 APPLE Etch Board **$21.00**
7510 APPLE Solder Tail Board **$21.00**
16AM APPLE, TRS-80 16K Add-On Memory **$120.00**

S-100 Bus Fare

2016 MXVI 16K Static RAM Board . . . **$299.00**
For expansion even beyond 64K . . . uses popular 2114 static RAMs, addressable in 4K blocks.

2520 Extender Terminator **$39.95**
With all lines labelled . . . power lines fused for added safety . . . active & dynamic termination . . . designed to eliminate crosstalk.

2501 Mother Board Kit With Active Terminations & Sockets **$119.95**

4P10 Four Parallel I/O Board **$99.95**

2500 Wire Wrap Board **$30.00**
Accommodates up to 102 16-pin sockets, or 24 40-pin sockets.

2590 Etch Board **$17.95**
For etching your own S-100 circuit . . . ½-oz. copper PC board . . . hard-gold-plate fingers.

2510 Solder Tail Board **$30.00**
Accommodates up to 102 16-pin sockets, or 24 40-pin sockets.

2200 All-Metal Mainframe Box **$399.00**
12-slot card cage with power supply and fan . . . inputs are 105, 115 and 125V AC . . . outputs are +8V at 20 amps and ±16V at 4 amps.

*APPLE II is a registered trademark of APPLE Computers, Inc.
‡TRS-80 is a registered trademark of Radio Shack, a Tandy Co.

Text continued from page 21:

At least part of the output $\mathbf{P}_4$ becomes part of the input command vector $\mathbf{C}_3$ to the next lower level. $\mathbf{C}_3$ is also a symbolic vector which identifies one of a library of *simple task* commands together with the necessary modifiers and arguments. As the feedback $\mathbf{F}_4$ varies with time, the input vector $\mathbf{S}_4$, and hence the output vector $\mathbf{P}_4$, move along a trajectory generating a sequence of simple task commands at $\mathbf{C}_3$ such as (FETCH A), (FETCH B), (MATE B TO A), (FASTEN B TO A), etc. as shown in figure 17b.

Feedback at $\mathbf{F}_3$ may identify the position and orientation of the parts A and B, and also carry state sequencing information from outputs $\mathbf{P}_3$ and $\mathbf{P}_2$. As $\mathbf{F}_3$ varies with time, it drives the input $\mathbf{S}_3$ (and hence $\mathbf{P}_3$) along a trajectory generating a sequence of *elemental movement* commands at $\mathbf{C}_2$ such as (REACH TO A), (GRASP), (MOVE TO C), (RELEASE), etc.

Feedback at $\mathbf{F}_2$ may contain information from proximity sensors indicating the fine positioning error between the fingers and the objects to be manipulated, together with state sequencing information derived from $\mathbf{P}_2$ and $\mathbf{P}_1$. The operator H_2 produces $\mathbf{P}_2$, which denotes the proper velocity vectors $\mathbf{C}_1$ for the manipulator hand in joint angle coordinates.

Feedback $\mathbf{F}_2$ also provides joint angle position data necessary for the coordinate transformations performed by H_2. $\mathbf{P}_2$ provides reference, or setpoint commands, $\mathbf{C}_1$ to the servomechanism operator H_1. $\mathbf{F}_1$ provides position, velocity, and force information for the traditional servocomputations. The output $\mathbf{P}_1$ is a set of drive signals to the actuators.

Feedback enters this hierarchy at every level. At the lowest levels, the feedback is unprocessed (or nearly so) and hence is fast acting with very short delays. At higher levels, feedback data passes through more and more stages of an ascending, sensory processing hierarchy. Feedback thus closes a real time control loop at each level in the hierarchy. The lower level loops are simple and fast acting. The higher level loops are more sophisticated and slower.

At each level the feedback vector $\mathbf{F}$ drives the output vector $\mathbf{P}$ along its trajectory. Thus, at each level of the hierarchy, the time rate of change of the output vector $\mathbf{P}_i$ will be of the same order of magnitude as the feedback vector $\mathbf{F}_i$, and considerably more rapid than the command vector $\mathbf{C}_i$. The result is that each stage of the behavior generating hierarchy effectively *decomposes* an input task represented by a slowly changing $\mathbf{C}_i$ into a string of subtasks represented by a more rapidly changing $\mathbf{P}_i$.

At this point we perhaps should emphasize that the difference in time rate of change of the vectors at various levels in the hierarchy does *not* imply that the H operators are computing slower at the higher levels than at the lower. We will, in fact, assume that every H operator transforms $\mathbf{S}$ into $\mathbf{P}$ with the same computational delay Δt at all levels of the hierarchy. That is:

$$\mathbf{P}_i\,(t) = H_i\,(\mathbf{S}_i\,(t-\Delta t)) \text{ or } \mathbf{P}_i^k = H_i\,(\mathbf{S}_i^{k-1})$$

at every level. The slower time rate of change of $\mathbf{P}$ vectors at the higher levels stems from the fact that the $\mathbf{F}$ vectors driving the higher levels convey information about events which occur less frequently. In some cases certain components of higher level $\mathbf{F}$ vectors may require the integration of informaion over long time intervals or the recognition of symbolic messages with long word lengths.

When we represent time explicitly as in figure 17, we can label the relatively straight segments of the $\mathbf{T}_c$ trajectories as tasks and subtasks. Transitions between the subtasks in a sequence correspond to abrupt changes in $\mathbf{T}_c$.

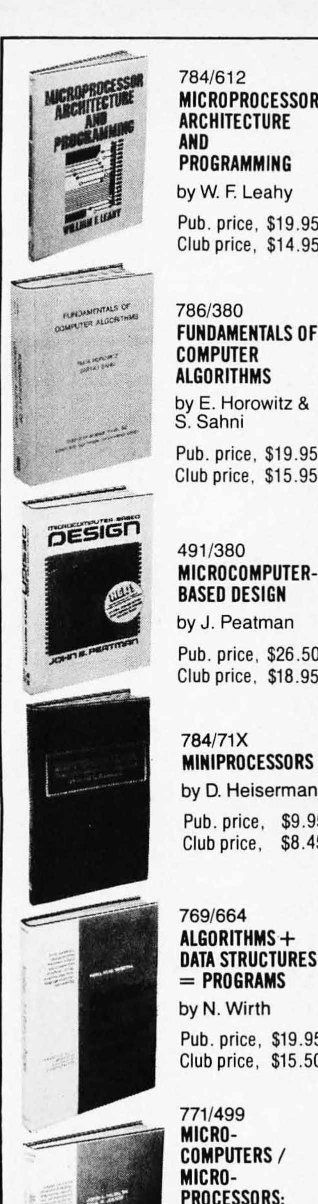
If we do not represent time explicitly, the relatively constant $\mathbf{C}$ vectors correspond to nodes, as in figure 15. The resulting tree structure represents a classical AND/OR decomposition of a task into sequences of subtasks, where the discrete $\mathbf{C}_i$ vectors correspond to OR nodes and the rapidly changing sequences of $\mathbf{P}_i$ vectors become sets of AND nodes *under* those OR nodes.

Intentional or Purposive Behavior

Figure 17 illustrates the power of a hierarchy of multivariant servomechanisms to generate a lengthy sequence of behavior which is both goal directed and appropriate to the environment. Such behavior appears to an external observer to be intentional, or purposive. The top level input command is a goal, or task, which is successively decomposed into subgoals, or subtasks, at each stage of the control hierarchy until at the lowest level output signals drive the muscles (or other actuators) producing observable behavior.

To the extent that the $\mathbf{F}$ vectors at the various levels contain sensory information from the environment, the task decompositions at those levels will be capable of responding to the environment. The type of response to each $\mathbf{F}$ vector depends on the H function at that level. If the $\mathbf{F}$ vector at any level is made up solely of internal variables,

Text continued on page 28



784/612
MICROPROCESSOR ARCHITECTURE AND PROGRAMMING
by W. F. Leahy
Pub. price, $19.95
Club price, $14.95

786/380
FUNDAMENTALS OF COMPUTER ALGORITHMS
by E. Horowitz & S. Sahni
Pub. price, $19.95
Club price, $15.95

491/380
MICROCOMPUTER-BASED DESIGN
by J. Peatman
Pub. price, $26.50
Club price, $18.95

784/71X
MINIPROCESSORS
by D. Heiserman
Pub. price, $9.95
Club price, $8.45

769/664
ALGORITHMS + DATA STRUCTURES = PROGRAMS
by N. Wirth
Pub. price, $19.95
Club price, $15.50

771/499
MICRO-COMPUTERS / MICRO-PROCESSORS: Hardware, Software and Applications
by J. L. Hilburn & P. N. Julich
Pub. price, $22.50
Club price, $16.50

Introductory offer to new members of the COMPUTER PROFESSIONALS' BOOK CLUB

any *two* of these great professional books for only $2.95 values up to $69.95

Two special bonus books come to you for $2.95 with your first club selection

Books are selected from a wide range of publishers

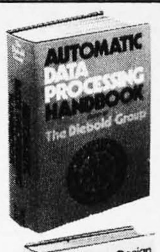
Be sure to consider these books as well—

784/701
DATA BASE SYSTEMS: Design, Implementation, and Management
by R. G. Ross
Pub. price, $19.95
Club price, $16.96

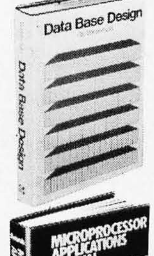
785/49X
PROGRAMMING FOR MINI-COMPUTERS
by J. C. Cluley
Pub. price, $17.50
Club price, $13.50

273/634
COMPUTER ARCHITECTURE & ORGANIZATION
by J. Hayes
Pub. price, $23.50
Club price, $17.50

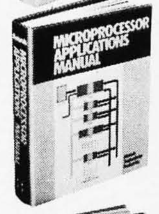
784/914
THE Z-80 MICRO-COMPUTER HANDBOOK
by W. Barden, Jr.
Pub. price, $8.95
Club price, $7.60



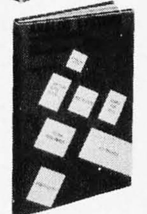
168/075
AUTOMATIC DATA PROCESSING HANDBOOK
Edited by The Diebold Group, Inc.
Pub. price, $34.95
Club price, $23.75



701/30X
DATA BASE DESIGN
by G. Wiederhold
Pub. price, $24.50
Club price, $18.25



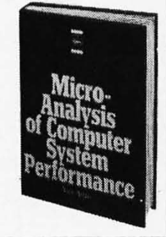
435/278
MICROPROCESSOR APPLICATIONS MANUAL
by Motorola, Inc.
Pub. price, $35.00
Club price, $25.00



786/496
ANALOG SYSTEMS FOR MICRO-PROCESSORS AND MINICOMPUTERS
by P. H. Garrett
Pub. price, $18.95
Club price, $14.95



783/845
THE 8080A BUGBOOK MICROCOMPUTER INTERFACING AND PROGRAMMING
by P. R. Rony, D. G. Larsen & J. A. Titus
Pub. price, $9.95
Club price, $8.45



785/058
MICRO-ANALYSIS OF COMPUTER SYSTEM PERFORMANCE
by B. Beizer
Pub. price, $22.50
Club price, $17.25

save time and money by joining McGraw-Hill's COMPUTER PROFESSIONALS' BOOK CLUB

McGraw-Hill

THIS professional club is designed to meet your day-to-day on-the-job needs by providing practical books in your field on a regular basis at *below publisher prices*. If you're missing out on important technical literature—if today's high cost of reading curbs the growth of your library—here's the solution to your problem.

The Computer Professionals' Book Club was organized for you, to provide an economical reading program that cannot fail to be of value. Administered by the McGraw-Hill Book Company, all books are chosen by qualified editors and consultants. Their understanding of the standards and values of the literature in your field guarantees the appropriateness of the selections.

How the Club operates: Thirteen times a year you receive free of charge *The Computer Professionals' Book Club Bulletin*. This announces and describes the Club's featured book of the month as well as alternate selections available at special members' prices. If you want to examine the Club's feature of the month, you do nothing. If you prefer one of the alternate selections—or if you want no book at all—you notify the Club by returning the card enclosed with each *Bulletin*.

As a Club Member, you agree only to the purchase of four books (including your first selection) over a two-year period. Considering the many books published annually, there will surely be at least four you would want to own anyway. By joining the club, you save both money and the trouble of searching for the best books.

MAIL THIS COUPON TODAY

COMPUTER PROFESSIONALS' Book Club
P.O. Box 582, Hightstown, New Jersey 08520

Please enroll me as a member and send me the three books indicated. I am to receive the two bonus books at the introductory price of $2.95 plus my first selection, plus tax, postage, and handling. If not completely satisfied, I may return the books within 10 days and request that my membership be cancelled. If I keep the books, I agree to take a minimum of three additional books during the next two years at special Club prices (guaranteed 15% discount, often more). I will receive the Club Bulletin 13 times a year. If I want to examine the featured selection, I need take no action. It will be shipped automatically. If, however, I want an alternate selection—or no book at all—I simply notify the Club by returning the convenient card always enclosed. I will always have a minimum of 10 days in which to return the card and you will credit my account fully, including postage, if this is not the case. Membership in the club is continuous but cancellable by me at any time after the four-book purchase requirement has been filled. *This order subject to acceptance by McGraw-Hill. Orders from outside the continental U.S. must be prepaid. Company, business, or institutional tax exemption status is not applicable to purchases made through individual Club memberships. All prices subject to change without notice. Offer good for new members only. A shipping and handling charge is added to all shipments. Members are billed when books arrive.*

Write Code # of 1st *bonus* book selection here	Write Code # of 2nd *bonus* book selection here	Write Code # of 1st book selection here

Name ______________________

Address ______________________

City ______________ State __________ Zip __________

EXTRA SAVINGS: Remit in full with your order, plus any local and state tax, and McGraw-Hill will pay all regular postage and handling charges.

P39373

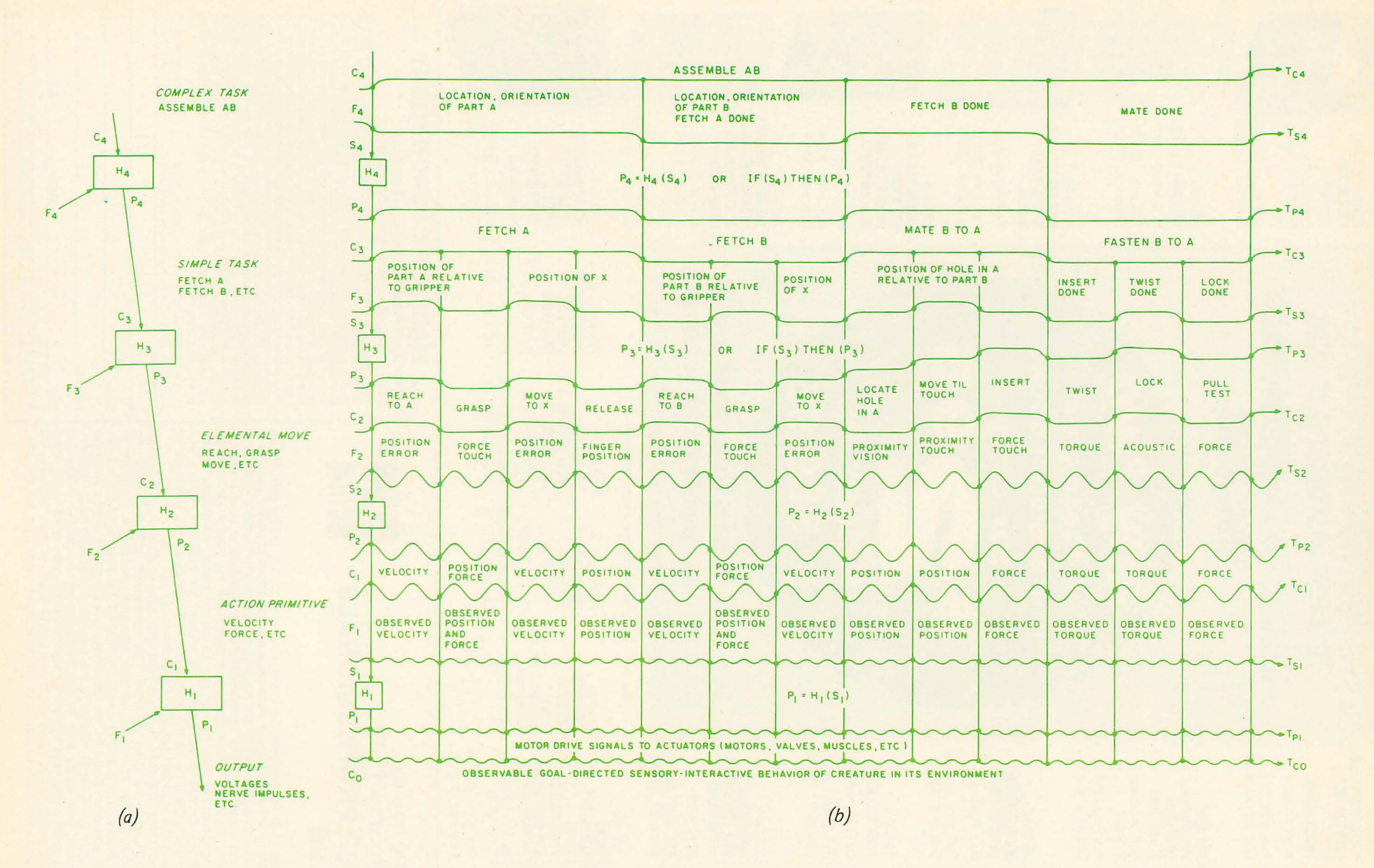


Figure 17: A hierarchy of H operators produces sensory interactive, goal directed behavior. The highest level input command C_4 *defines a goal, which in this example is (ASSEMBLE AB). The feedback* F_4 *carries highly processed sensory data describing the state of environment in which the assemble command must operate, including the state of the lower level* P *vectors. The* H_4 *operator maps each input* S_4 *into an output* P_4 *As* F_4 *changes the goal (ASSEMBLE AB) is decomposed into a sequence of subgoals (FETCH A), (FETCH B), (MATE B to A), (FASTEN B to A). At each level in the hierarchy a different type of feedback data with a different time-rate-of-change drives the decomposition of a higher level command into a sequence of lower level subcommands. Finally, at the lowest level the* P_0 *vector consists of motor drive signals which actuate observable behavior* C_0.

I've finally found a personal computer I respect. Compucolor II.

It's not surprising that professionals get excited about the Compucolor II. It's a totally-integrated 8080A system with full color graphics display, built-in 51K mini-disk drive, and the best cost performance ratio available in a personal computer.

The complete system is only $1495.* And that price includes 8K user RAM, RS-232C compatibility and random access file capabilities.

Our 8 foreground and background colors will boost your comprehension, while introducing you to an exciting new dimension in BASIC programming. The vector graphics have 16,484 individually-accessible plot blocks. And the 13" diagonal measure screen gives you 32 lines of 64 ASCII characters. You also have the flexibility that comes with 16K Extended Disk BASIC ROM.

Compucolor II offers a number of other options and accessories, like a second disk drive and expanded keyboard, as well as expandability to 32K of user RAM. Of course we also have a whole library of low-cost Sof-Disk™ programs, including an assembler and text editor.



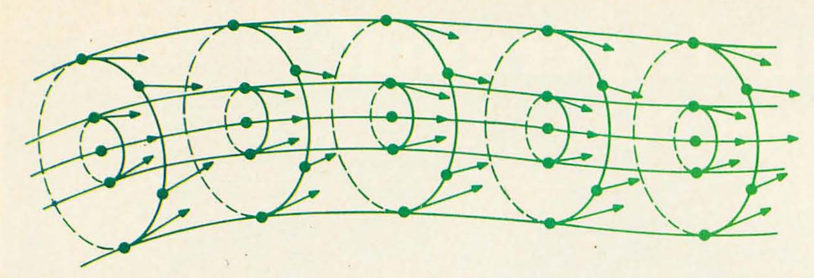


Figure 18: Around each trajectory representing an ideal task performance there exists an envelope of nearly ideal trajectories which correspond to successful, but not perfect, task performance. If the H functions are defined throughout these envelopes so as to drive the system back toward the ideal whenever it deviates, then the trajectory will be stable and task performance can be successful despite perturbations and unexpected events.

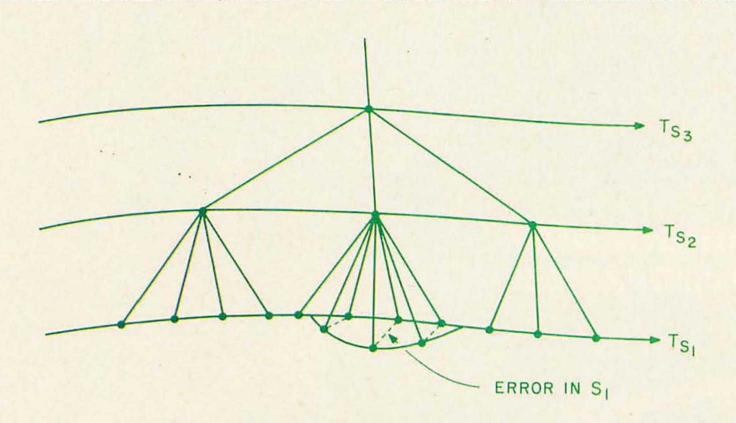


Figure 19: If the H functions at the lower levels are sufficiently well defined, small perturbations from the ideal performance can be corrected by low level feedback without requiring any change in the command from higher levels.

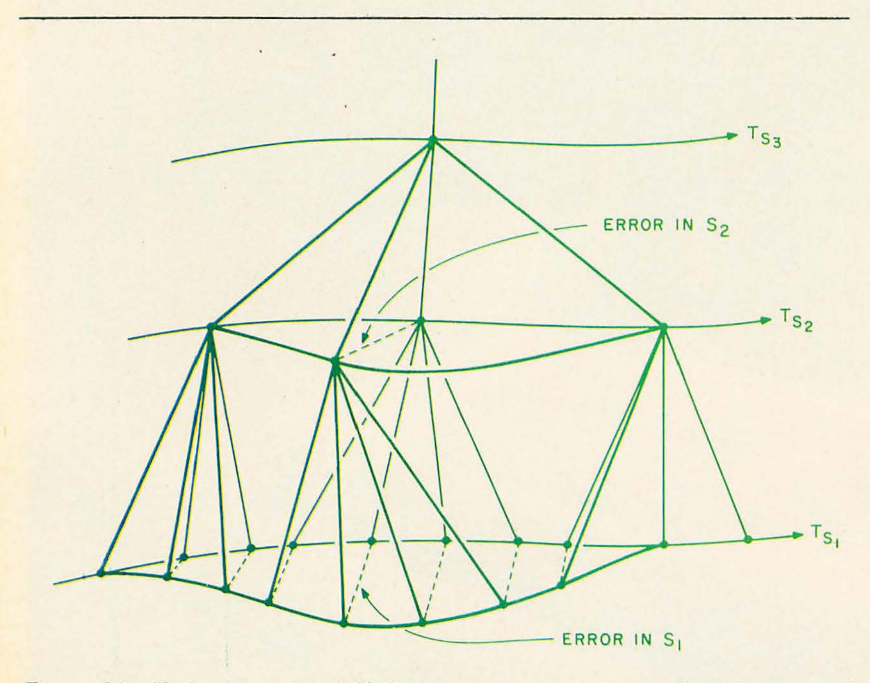


Figure 20: If the lower level H functions are not adequately defined, or if the perturbations are too large for the lower level to cope, then feedback to the higher levels produces changes in the task decomposition at a higher level. The result is an alternative strategy.

Text continued from page 24:
then the decomposition at that level will be stereotyped and insensitive to conditions in the environment.

Whether or not the hierarchy is driven by external or internal variables, or both, the highest level input command commits the entire structure to an organized and coordinated sequence of actions which under normal conditions will achieve the goal or accomplish the task. The selection of a high level input command in a biological organism thus corresponds to an intent or purpose, which, depending on circumstances, may or may not be successfully achieved through the resulting hierarchical decomposition into action.

Obtaining Successful Performance

The success or failure of any particular task performance, or goal seeking action, depends on whether or not the H functions at each level are capable of providing the correct mappings so as to maintain the output trajectory within a region of successful performance, despite perturbations and uncertainties in the environment.

At all levels, variations in the **F** vectors due to irregularities in the environment cause $\mathbf{T}_s$ trajectories to vary from one task performance to the next. This implies that while there may exist a set of ideal trajectories through **S** and **P** space at each level of the hierarchy corresponding to an ideal task performance, there also must be an envelope of nearly ideal trajectories which correspond to successful, but not perfect, task performance. This is illustrated in figure 18.

The H functions must not only be defined along the $\mathbf{T}_s$ trajectories corresponding to ideal performance, but also in the regions around the ideal performance. Consequently, any deviation from the ideal is treated as an error signal which generates an action designed to restore the actual trajectory to the ideal, or at least to maintain it within the region of successful performance.

Small perturbations can usually be corrected by low level feedback loops, as shown in figure 19. These involve relatively little sensory data processing, and hence are fast acting. Larger perturbations in the environment may overwhelm the lower level feedback loops, and require strategy changes at higher levels in order to maintain the system within the region of successful performance. This is illustrated in figure 20. Major changes in the environment are detected at higher levels after being processed through several levels of pattern recognizers. This produces differences in the **F** vector at the higher level

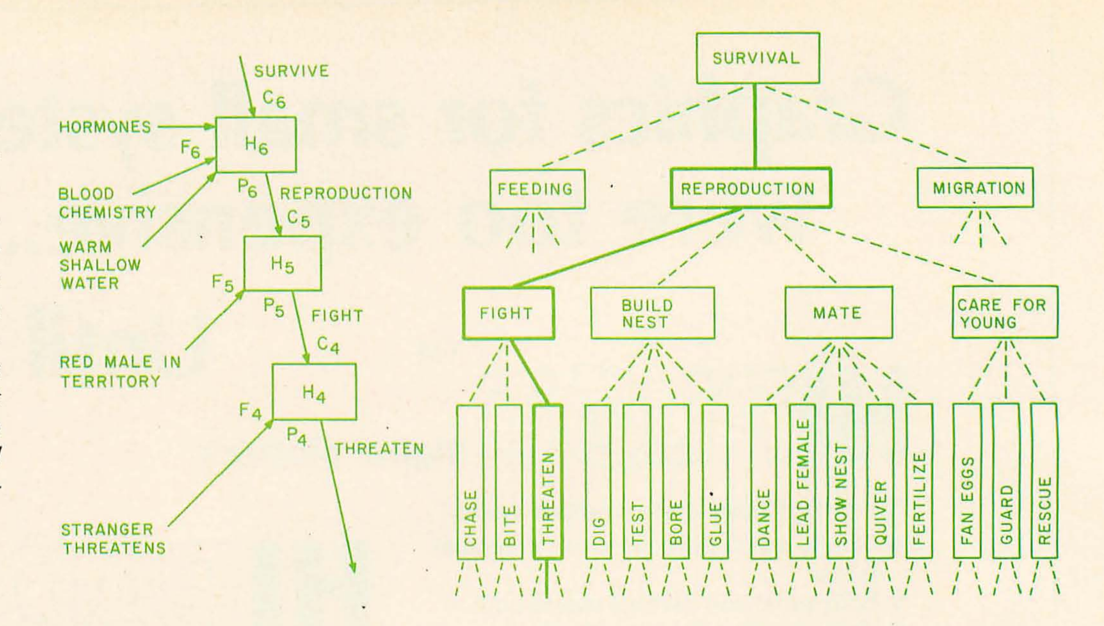


Figure 21: The command and control hierarchy proposed by Tinbergen to account for the behavior of the male 3 spined stickleback fish. The heavy line indicates the particular type of behavior vector actually selected by the feedback shown at the various levels of the hierarchy on the left. This figure represents a snapshot in time corresponding to one of the two-dimensional surfaces shown in figure 16.

which in turn produces different **C** vectors to lower levels. The result is an alternative higher level strategy to cope with the perturbation.

Of course, if the H functions do not provide stability, or if the environment is so perverse that the system is overwhelmed, then the trajectories diverge from the region of successful performance and failure occurs.

Over-learned tasks correspond to those for which the H functions at the lower levels are sufficiently well defined over a large enough region of input space so as to maintain the terminal trajectory well within regions of stability and success without requiring intervention by the higher levels for strategy modification. Thus, a highly skilled and well-practiced performer, such as a water skier, can execute extremely difficult maneuvers with apparent ease despite large perturbations such as waves. His lower level H functions are well defined over large regions of space corresponding to large perturbations in the environment. He is thus capable of compensating for these perturbations quickly and precisely so as to maintain successful performance without intervention by higher levels. Such a performance is characterized by a minimum amount of physical and mental effort.

We say, "He skis effortlessly without even thinking." What we mean is that his lower level corrections are so quick and precise that his performance never deviates significantly from the ideal. There is never any need for higher level loops to make emergency changes in strategy. On the other hand, a novice skier (whose H functions are poorly defined, even near the ideal trajectory, and completely undefined elsewhere) may have great difficulty maintaining a successful performance at all. He is continually forced to bring higher levels into play to prevent failure, and even the slightest perturbation from the ideal is likely to result in a watery catastrophe. He works very hard, and fails often, because his responses are late and often misdirected. His performance is erratic and hardly ever near the ideal.

However, practice makes perfect, at least in creatures with the capacity to learn. Each time a trajectory is traversed, if there is some way of knowing what mistakes were made, corrections can be made to the H functions in those regions of input spaces which are traversed. The degree and precision of these corrections, and the algorithm by which they are computed, determine the rate of convergence (if any) of the learning process to a stable and efficient success trajectory.

There are many interesting questions about learning, generalization, and the mechanisms by which H functions are created and modified at the various hierarchical levels in biological brains. However, we will defer these issues until part 2 (July 1979 BYTE).

Task Decomposition and Goal Seeking

Note that figure 17 illustrates only a single specific performance of a particular task. None of the alternative trajectories which might have occurred under different circumstances with a different set of F vectors are indicated. These alternatives which might have occurred can be illustrated in the plane orthogonal to the time axis.

Figure 21 illustrates the set of alternative C vectors available at various levels in the behavior-generating hierarchy of the male 3 spined stickleback fish. This figure

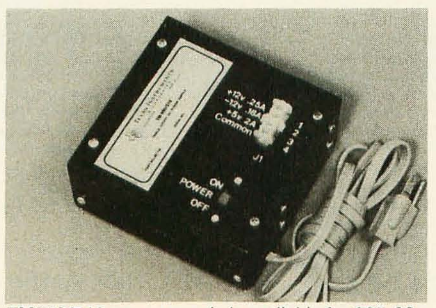
9900

New easy-to-use educational module. Your best ticket to the microprocessor world. Ideal for teaching. Yourself. Or others.

TI's new TM990/189 University Module is a stand-alone learning lab. Fully assembled and designed for maximum hands-on experience. To ease and simplify learning and teaching.

Outstanding features include powerful 16-bit microprocessor with easy-to-learn, easy-to-use minicomputer instruction set; 45-key alphanumeric keyboard and ten-digit seven-segment display for easy **assembly-language** programming; ROM-resident software including system monitor for program debug and symbolic assembler; audio cassette interface; easy-to-add EIA and TTY interface; 1K-byte RAM expandable to 2K; 4K-byte ROM and 2K-byte expansion EPROM socket; 16-bit programmable I/O controller; user addressable LEDs, and sound indicator.

A **570-page tutorial text** accompanies the module. It is a detailed guide for self-paced learning. Or the basis for a three-hour university course. Chapters include an overview of microprocessors; programming exercises; assembly language; memory systems; I/O concepts and designs; software engineering; product development; a variety of lab experiments, and much more. Also with the module: a 300-page user's guide.



TM990/519 power supply is available for $65.00.

The University Module complete with tutorial text and user's guide is only $299.00.

Order your University Module from your authorized TI distributor. For more information, write Texas Instruments Incorporated, P.O. Box 1443, M/S 6404, Houston, Texas 77001.

TEXAS INSTRUMENTS MOVING AHEAD IN MICROCOMPUTERS



Circle 266 on inquiry card.

TEXAS INSTRUMENTS
INCORPORATED

represents a snapshot, or single cut through space orthogonal to the time axis. C_4, the highest level goal, is *survival*. The feedback F_4 consists of variables indicating water temperature and depth, blood chemistry, and hormone levels generated by length of day detectors. When the hormone levels indicate the proper time of year and the blood chemistry does not call for feeding behavior, then migratory behavior will be selected until warm, shallow water is detected. The F_4 vector will then trigger the reproduction subgoal.

When C_3 indicates (REPRODUCTION), an F_3 vector indicating a red male in the territory will cause the (FIGHT) command to be selected to C_2. When C_2 indicates (FIGHT) and the intruder threatens, a C_1 will be selected, and so on. At each level, a different feedback vector would select a different lower level subgoal. For example, if F_3 indicates a female in the territory, C_2 will become (MATE), and the type of mating behavior selected will depend on F_2.

In simple creatures like the stickleback fish, the sensory stimuli that produce F_2 and F_3 vectors which trigger specific behavioral trajectories are called *innate releasing mechanisms*. Innate releasing mechanisms and their associated behavioral patterns have been studied extensively in a number of insects (ie: the digger wasp and various bee and ant species), several fish, and many birds (ie: the herring gull, turkey, and golden eye drake).

In these relatively simple creatures, behavior is sufficiently stereotyped that it can be described in terms of a small set of behavioral patterns triggered by an equally small set of sensory stimuli. This suggests that insects, fish, and birds have only a few levels in their control hierarchies and a small set of behavior patterns stored as H functions at each level. It further implies that there are few externally driven components in the F vectors at each level. Behavior trajectories are internally driven, with only a few branch points controlled by sensory data processed through simple pattern recognizers. The trajectory segments driven entirely by internal variables are called fixed action patterns, or *tropisms*. The external variables which control the relatively few branch points are the innate releasing mechanisms.

In higher animals, behavior is more complex and much less stereotyped. This implies more levels in the hierarchy, more external sensory variables in the F vectors at each level, and hence many more possibilities for branching of the resulting trajectories.

Figure 22 illustrates a set of trajectories

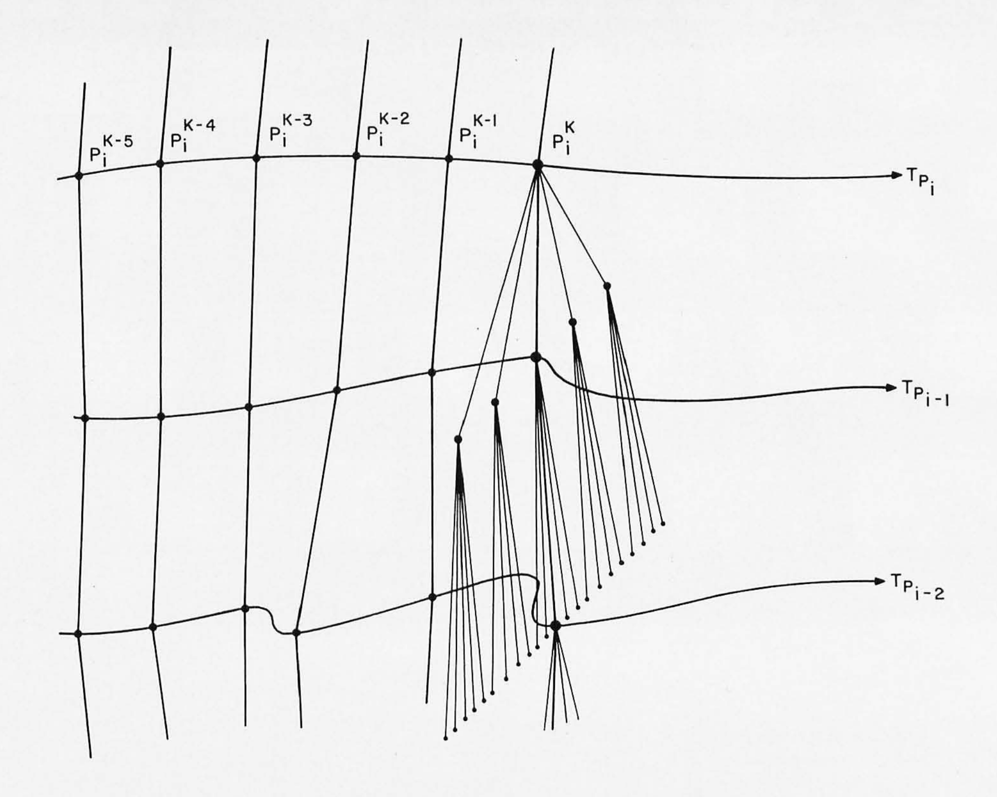


Figure 22: A set of $\mathbf{T}_p$ *trajectories in which there is opportunity for branching at many points in time. If behavior can be modified by feedback at many different levels and in many different ways, it appears to be adaptive and flexible. If there are only a few branch points, with only a few alternative actions available at each branch, behavior will appear stereotyped.*

in which there is opportunity for branching at *several* different levels at *every* step along each trajectory. At each instant in time the **C** vector to any particular level depends upon what the **C** and **F** vectors were to the next higher level at the previous instant. Thus, a change in the **F** vector at any level causes an alternative **C** vector to be sent to the level below. Behavior is continuously modified at all levels by external variables, and hence does not appear stereotyped at all.

Many degrees of freedom place great demands on the H functions for maintaining stability and precision of control in such a large space of possibilities. Since successful behavior is only a tiny subset of all possible behaviors, it is clear that most of the potential branches will lead to disaster unless the H functions produce actions which steer the **S** and **P** vectors back into the narrow regions surrounding success trajectories. For a multilevel hierarchy with sensory interaction at many different levels, this is extremely complex. However, if the H functions are trainable, then performance can improve through practice. Complex tasks can be learned, imitated, and communicated from one individual to another.

Conclusion

We have now completed the first step in our development. We have described a hierarchical computing structure which can execute goals, or intended tasks, in an unpredictable environment. We have also defined a notation by which the behavior of such a hierarchy can be described clearly and concisely. We have asserted that the complexity of behavior resulting from such a control hierarchy depends on four things:

- the number of levels in the control hierarchy;
- the number of feedback variables which enter each level;
- the sophistication of the H functions which reside at each level;
- the sophistication of the sensory processing systems which extract feedback variables for use by the various H functions.

In part 2 we will describe a computer model of a neurophysiological structure in the brain which computes multivariant H functions. We will then suggest how the brain might use such structures to learn skills, remember events, select goals, and plan future actions.■

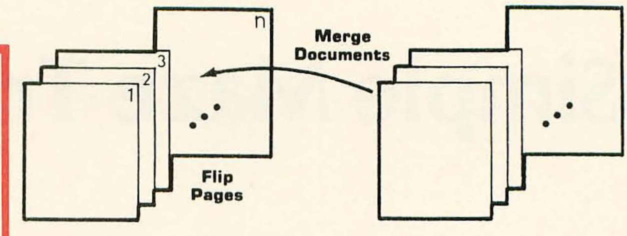
BIBLIOGRAPHY

1. Albus, J S and J M Evans, "Robot Systems," *Scientific American,* February 1976, pages 77 thru 87.
2. Arbib, M A, *The Metaphorical Brain: An Introduction to Cybernetics as Artificial Intelligence and Brain Theory,* Wiley-Interscience, New York, 1972.
3. Sommerhoff, G, *Logic of the Living Brain,* Wiley-Interscience, New York, 1974.
4. Tinbergen, N, *The Study of Instinct,* Clarendon Press, Oxford, 1951.

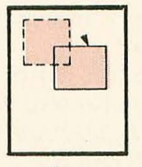
Wordsmith is the video text editing system you've been waiting for. Its power, flexibility and simplicity help you carve any text editing task down to size—in a way you can understand. We wanted a system that allows you to think in traditional ways about text layout, yet at the same time makes the traditionally tedious operations such as cut and paste simple and fast. We think we've done it. We want you to decide for yourself.

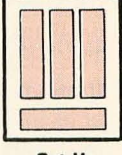
Power

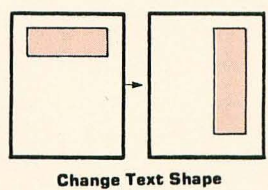
- **Page Oriented Philosophy.** A document is a collection of pages. The screen displays one entire page at a time. Simple random access page flipping commands take you quickly to any page in the document. Equally efficient commands allow you to insert, delete, copy and move pages both within one document and across documents.



- **Extensive Block Manipulation Capabilities.** Using "windows", portions of text, charts, etc., can be quickly and effortlessly moved around on the current page, or across pages. The shape and size of any window can be changed in real time, with the contained text automatically reformatting itself (heeding word and paragraph boundaries) to conform to the new shape.


Move Text Blocks

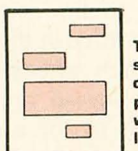

Set Up Multiple Text Regions

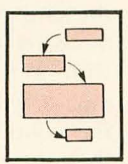

Change Text Shape

- **Instantaneous Formatting.** Compacting (extraneous blank deletion) and right justifying are simple commands that tidy up a full page or window's worth of text in the blink of an eye. Random access cursor movement, line and character insert and delete, line and page split and join, and a host of other line and character level commands help you put text in its place quickly and accurately.

Flexibility

- **Logical/Physical Page Distinction.** Define your own hardcopy size. Wordsmith remembers the difference between the screen size and the hardcopy page size.
- **Modular Hardcopy Driver.** Drive a Qume® Sprint-5 or TTY-like device directly now, Diablo, NEC and other hardcopy devices soon.
- **Pure Text.** Wordsmith files are pure text with no control characters mixed in. This universal format keeps you as compatible with the world as possible. What you see on the screen is what you get as hardcopy.
- **Page Templates.** Snapshots of the block layout of a page can be saved as named disk files, then later recalled and superimposed on the current page. Use such "templates" for standard multicolumn layouts, common letter formats, and fixed-field forms. A single keystroke dispatches you quickly from block to block as you fill in your page.

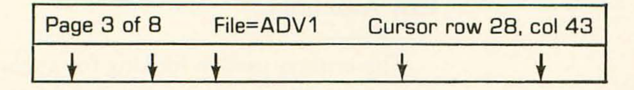

Take a snapshot of a page's window layout


Recall a template and hop from window to window.

- **File Switching.** Moving from document to document to examine, copy, move and change text is like rolling off a log. You're not confined to one disk file at a time anymore.

Simplicity

- **Auto Word Break.** Forget the right margin. Wordsmith notices when you won't be able to complete the current word and moves it to the next line for you as you continue typing.
- **Understandable Commands.** The most frequently used commands are single keystrokes. The rest are easily remembered abbreviations.
- **Informative Status Lines.** The top two screen lines constantly display page number information, document name, cursor position, tab stops and status/error phrases. You're always in touch with your document.



- **Protection Against Catastrophic Errors.** It's nearly impossible to ruin your document with a single bad command. Wordsmith's page oriented design and double-checking user interface help you do what you mean!

The Wordsmith™ TEXT EDITOR

Defining the New Generation of Text Editing

from Micro Diversions, Inc.
8455-D Tyco Rd.
Vienna, Va. 22180
(703) 827-0888

- Direct CP/M® and North Star DOS compatibility
- Available for 40x86, 24x80 and 16x64 memory-mapped video boards
- Fully reentrant for efficient multi-programming environments (6K program space, 5K data area)
- 8080 and Z80 compatibility

Ordering Information:
$200
(Screensplitter™ Owners: $80)
Manual only: $15
Check, VISA, Mastercharge

1. CP/M or North Star DOS version?
2. TTY or QUME interface?
3. Brand and memory address of video display board?
4. Ship on single or double density, 5" or 8" diskette?

Inquire about our custom keyboard.

Simple Maze Traversal Algorithms

Sandra and Stephen A Allen
2543 Luciernaga St
Carlsbad CA 92008

This article is a discussion of some solutions to the problem of traversing a maze. The algorithms here represent neither an exhaustive nor a high-powered study of the problem, but rather an intuitive approach. The ideas reflect our thought processes and those of Tony Rossetti in our efforts to compete in *IEEE Spectrum* magazine's ongoing Micromouse contest.

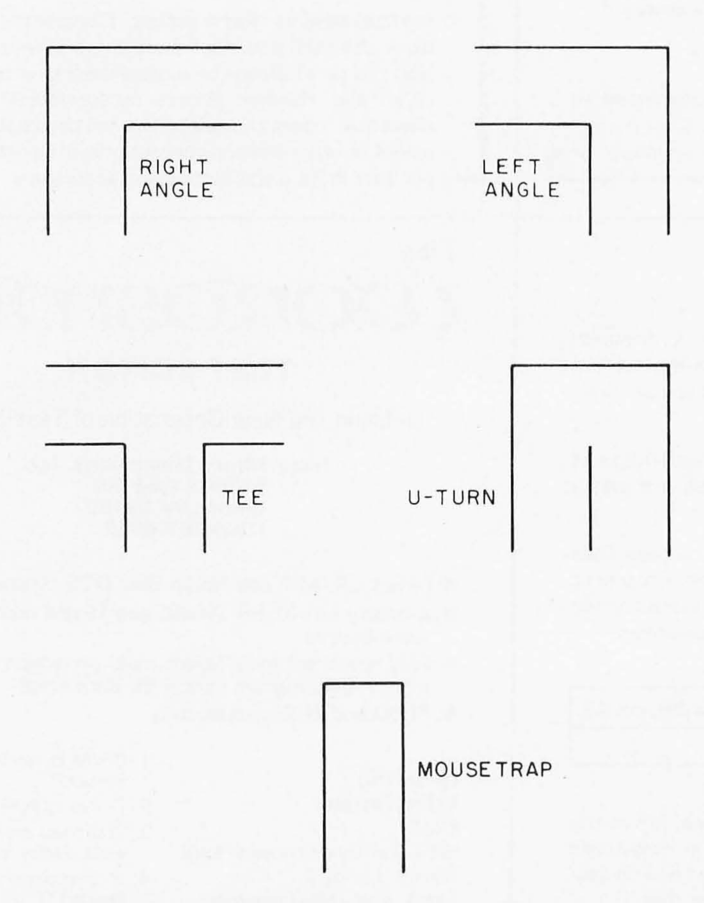


Figure 1: Types of intersections allowed in the maze. All intersections are at right angles and no cross intersections are allowed.

Problem Specification

The *IEEE Spectrum*'s Micromouse Maze contest began time trials last June at the 1978 National Computer Conference in Anaheim CA. A mechanical "mouse" (ie: robot) must find its way under its own power from the entrance of a maze to the exit. Each mouse is given three tries through the maze, with a time limit on each attempt. The mouse with the shortest logged time wins the contest. The solution to the problem, then, is to find the path through the maze that yields the shortest time.

An important consideration in finding a solution is the characteristics of the maze. The corridors are of uniform width. There are only five types of intersections: right angle; left angle; T; U turn; and mousetrap. These are shown in figure 1. There are no cross-intersections, nor are there any king's chambers, which are large vacant areas in the maze. This simplifies the traversal algorithms somewhat. Finally, there are exactly one entrance and one exit on the perimeter of the maze, but not necessarily on opposite sides.

Characteristics of the mouse should also be taken into consideration. It must be completely self-contained, having an on board computer and any required memory. Since the mouse must carry its own battery, available power is a limiting factor.

Easy Algorithms

The criteria used in looking for a solution were primarily based on the considerations discussed above. The limited power, program space, memory and processor power were perhaps the most important aspects. Simplicity was also an important element in order to provide easy modification and enhancement of the robot.

The traversal algorithm that suggests itself first is the simplest one: each time the robot encounters an intersection, the rightmost path is followed. This is the equivalent of the robot resting its "right hand" on the right wall of the maze corridor and not lifting it throughout the maze walk (see figure 2). Of course it works in the same way when taking the leftmost path instead of the rightmost. This algorithm certainly meets the requirement of simplicity, and the program is small and uses little memory. It also guarantees a solution, but unfortunately in using this algorithm there only exists one solution, so there is no way to take advantage of the second and third runs through the maze to significantly improve the traversal time.

The resulting path may be a very long, roundabout path through the maze. This, of course, depends on the configuration of the particular maze. Indeed, the possibility exists that the path takes more than the allotted time for the robot to complete. So the leftmost/rightmost algorithm guarantees one solution, but has significant disadvantages.

A variation of this algorithm is one in which, at each intersection, the robot chooses one of the available paths at random. Again, this is an easy and short program to implement. It has the important advantage that a different path is picked every time through the maze, so that there is the chance of picking a fairly short path. A definite drawback, though, is that no solution is guaranteed. The robot may come out the entrance instead of the exit, or may wander around in the maze until the allotted time has run out.

Another slight variation on the leftmost/rightmost algorithm is the *straight-ahead first* algorithm. If, in this algorithm, a straight-ahead alternative exists at an intersection, that path is chosen (otherwise either the right or left path is taken). Like the former algorithm, this one is short and simple, yielding a unique path; however, this path may not be a solution. This algorithm can have certain advantages, though. When a maze has cross intersections and the entrance and exit are on opposite sides of the maze, the robot gets close to the exit faster. Also, for robots which are mechanically faster at going straight than at turning, this algorithm can minimize the required turns.

All of the solutions discussed above are very simple and, except for the random path algorithm, give only one possible solution for each maze configuration. Since the mouse gets three tries through the maze, a logical next step is to use a different algorithm on each run. For example, choose the rightmost path on the first time through, the

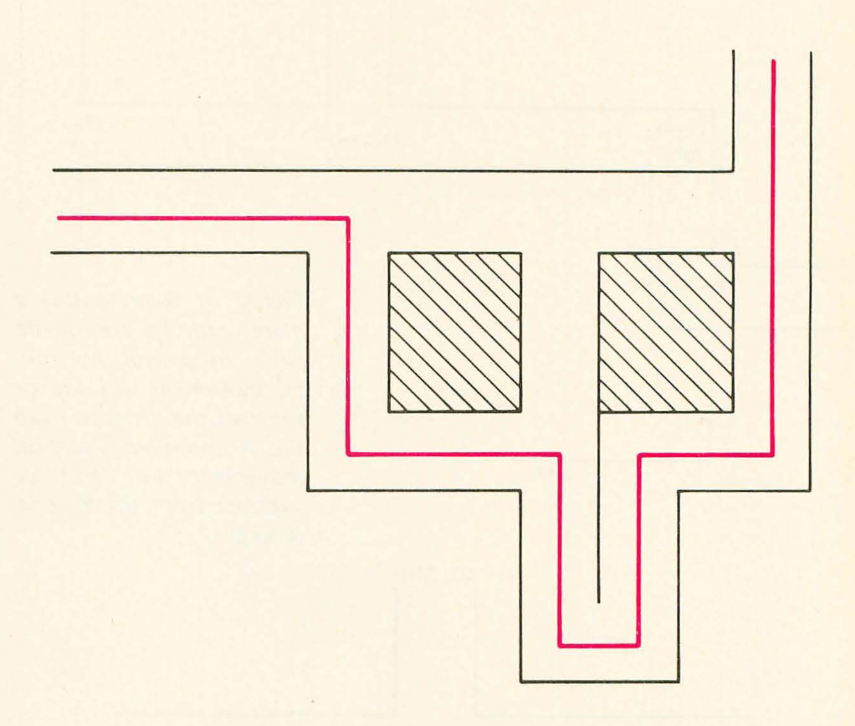


Figure 2: Example of a rightmost, maze walk. At each intersection encountered, the rightmost available path is chosen.

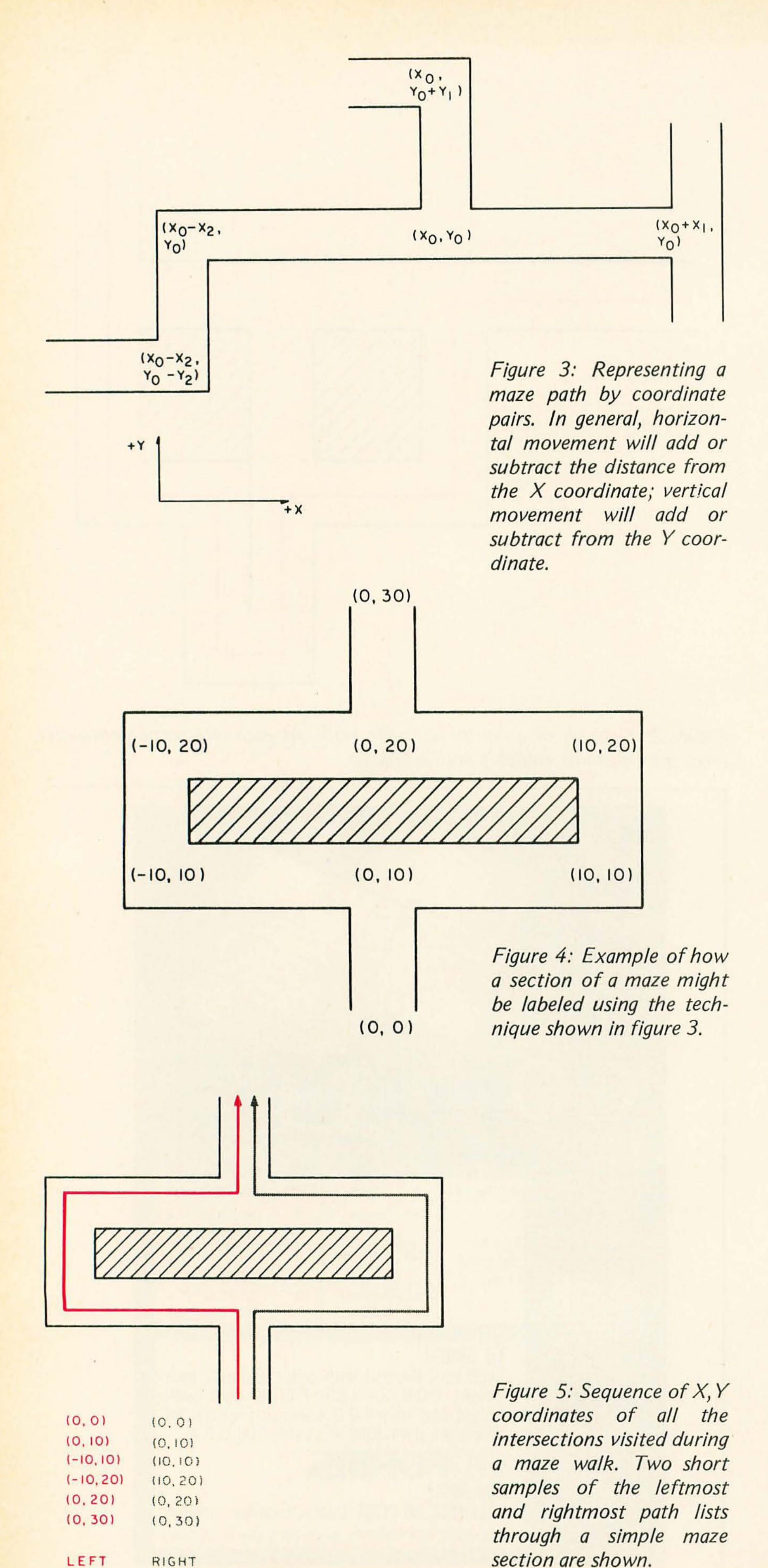


Figure 3: Representing a maze path by coordinate pairs. In general, horizontal movement will add or subtract the distance from the X coordinate; vertical movement will add or subtract from the Y coordinate.

Figure 4: Example of how a section of a maze might be labeled using the technique shown in figure 3.

Figure 5: Sequence of X,Y coordinates of all the intersections visited during a maze walk. Two short samples of the leftmost and rightmost path lists through a simple maze section are shown.

leftmost path the second time, and the straightest path the third time. In this way the mouse is allowed to take advantage of having three tries, and perhaps one will yield a reasonably short time.

Smarter Algorithms

An unfortunate characteristic of mazes built for a competition of this sort is that wall-hugging mice (ie: those that don't recognize corridors and intersections, but are built to blindly follow the right or left wall) are heavily penalized. This means that the maze probably has a fairly short and direct path from the entrance to the exit, but that this path has so many dead-end offshoots that a wall-hugger ends up covering a large portion of the maze's interior before reaching the exit. This aspect makes it not such a good idea to use only a rightmost or leftmost algorithm. Rather, it argues for using an algorithm that can be smart about picking the path and learning from its mistakes.

Obviously, the only way the robot can learn from its mistakes is by remembering what it did. In this way, the robot can make a first try through the maze using one of the simple algorithms discussed before, remembering the path taken. Then the remembered path can be optimized. This attack gives the potential for significant time improvement.

Remembering the Path Taken

The most difficult part of remembering a path taken through the maze is how to represent that path in the mouse's memory. One straightforward way to do this is to conceptually map the maze onto an X,Y coordinate grid, picking a convenient origin (ie: the entrance to the maze) and orientation (ie: forward from the entrance is +X). Since all intersections of corridors are at right angles, any movement will be either parallel to the X axis or parallel to the Y axis, and will have either a positive or negative increment. This mechanism provides a way to uniquely name all intersections in a maze simply by giving the X,Y pair which specifies the distances from the entrance at (0,0). See figures 3 and 4 for examples of naming intersections.

As the mouse visits intersections on its walk through the maze, it can record in its memory the X,Y coordinates of each, thus generating a list of X,Y pairs starting with (0,0) which uniquely describes the particular path taken. Figure 5 shows the coordinate lists representing the leftmost and rightmost paths through the labeled maze of figure 4.

ELEPHANT MEMORY

for peanuts!

At last, you can get a mammoth 32K memory without paying a beastly price. The SuperRam™ 32K **static** memory kit is just $649 for S-100 systems.

The latest of George Morrow's heavyweight memory designs for Thinker Toys™, the SuperRam™ 32K features two independent 16K blocks, each addressable and write-protectable. It meets the Proposed IEEE S-100 Standard with full buffering of both data and address lines. Uncommonly efficient, the Morrow design utilizes just 7 IC's for control and buffering, drawing typically 2.6 amps.

Ask your local computer shop to order the SuperRam™ 32K kit for you. Or call (415) 524-2101.

Morrow makes memory for

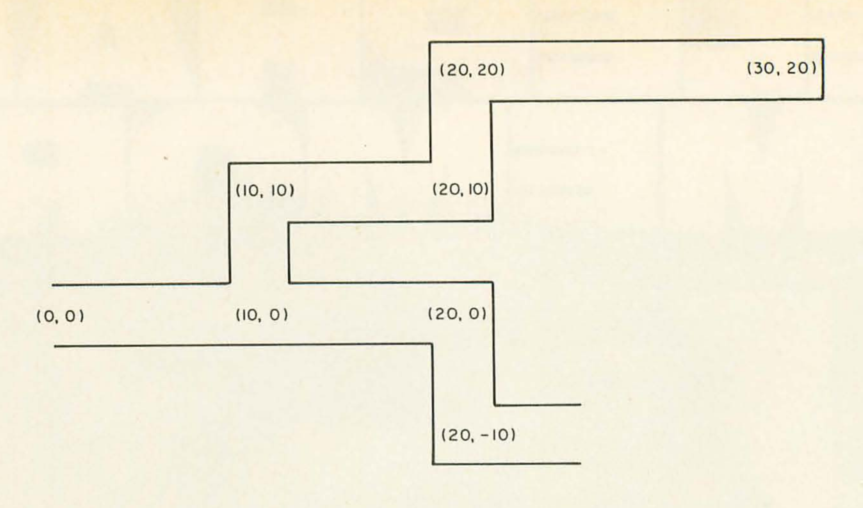


5221 Central Ave., Richmond, CA 94804

"Judy" the elephant courtesy Marine World—Africa USA

Circle 256 on inquiry card.

"See us in booths 98 and 99 at the NCC"



LEFTMOST PATH BEFORE OPTIMIZATION	LEFTMOST PATH AFTER OPTIMIZATION
(0, 0)	(0, 0)
(10, 0)	(10, 0)
(10, 10)	(20, 0)
(20, 10)	(20, -10)
(20, 20)	
(30, 20)	
(20, 20)	
(20, 10)	
(10, 10)	
(10, 0)	
(20, 0)	
(20, -10)	

Figure 6: Maze pitfalls. On its first, nonoptimized attempt, the mouse will traverse the entire length of the dead-end corridor. After optimizing the path by deleting backtracked corridors, a shorter path is taken.

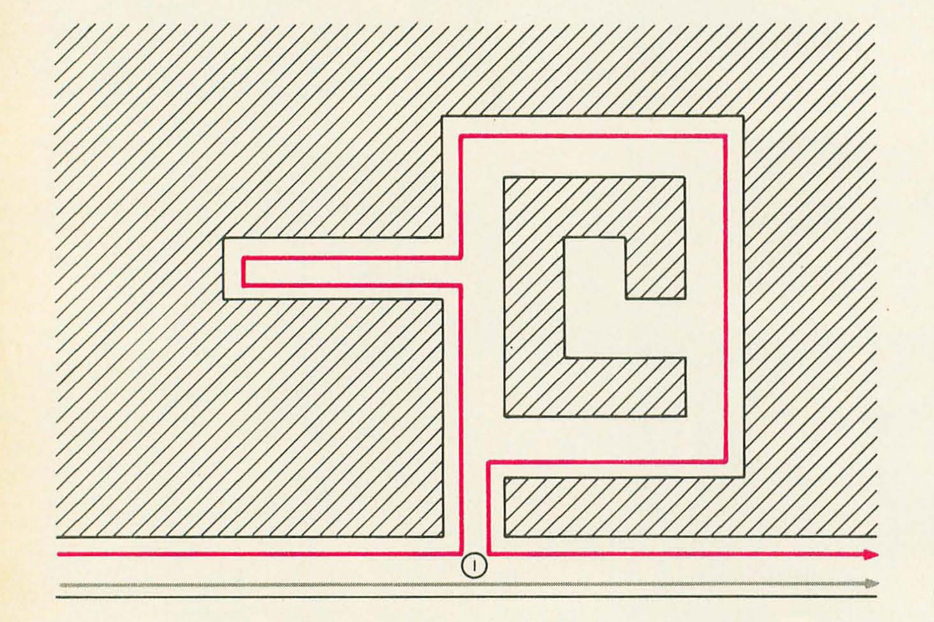


Figure 7: Leftmost versus rightmost maze traversal, in which the robot keeps its right or left side against the respective wall while traversing the maze. The leftmost path the mouse would travel is shown in red. Notice that the outer edge of the loop is completely circumvented. When intersection 1 is visited a second time it is clear that this whole section of the maze is fruitless because all corridors leading away from the center of the loop were tried and found to be useless. Since the exit to the maze is along the perimeter of the maze, no corridor going to the inside of the loop could possibly reach the exit. The rightmost path is shown in gray: in this case the whole loop is neatly bypassed. The mirror image of the above maze section can be used to argue similarly for chopping the loop off the rightmost path.

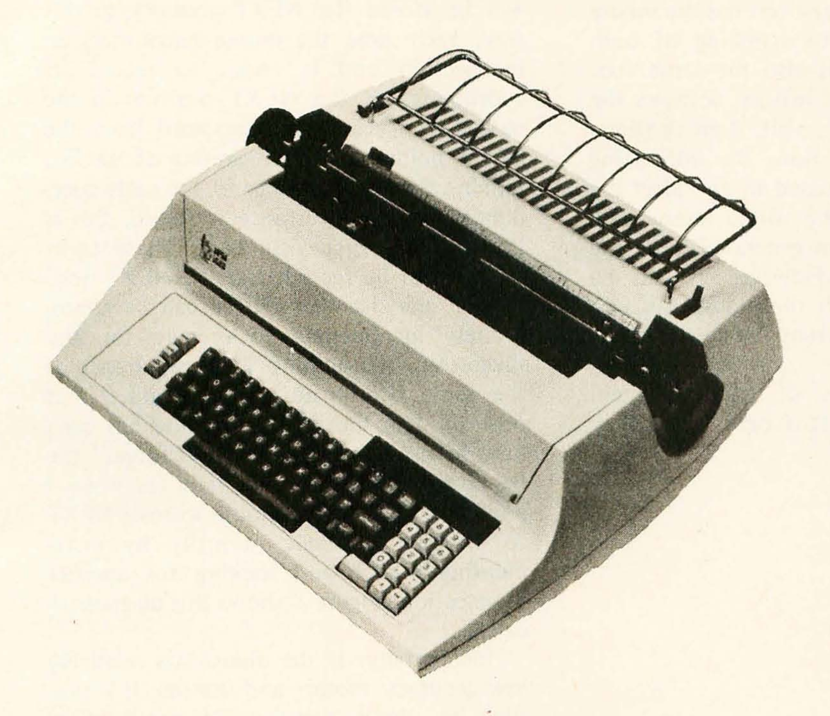
Optimization

Now that the mouse has a way of remembering the path it takes on the first attempt through the maze, the next problem is, how can it find a better (shorter) path for the second attempt? The main motivation behind optimizing a path is to chop off all parts of the path that don't contribute to getting closer to the exit. That is, remove any part that had to be backtracked.

The first type of backtracking is that in which a corridor is a dead end, and the mouse has to return to the main corridor to continue. Clearly the time spent negotiating this part of the maze is wasted and should be omitted. Figure 6 illustrates this kind of backtrack optimization: every coordinate pair along the backtracked path can be deleted from the list. Notice that this same method works no matter how long the dead end corridor is (ie: how many intersections it has).

A second kind of backtracking occurs when the maze contains a loop or cycle. The presence of a loop is indicated when the mouse returns to an intersection that it has previously visited. The whole traversal of the loop can be cut out of the maze walk since that part of the maze is useless and cannot lead to the exit. Figure 7 illustrates how it is always the outer edge of the loop that is traversed. Once an intersection is revisited, all corridors leading away from the center of the loop have been already tried and found to be fruitless (otherwise the mouse would not have returned to the loop entrance). And since the exit is along the perimeter of the maze, no corridor leading to the inside of the loop could

NOW YOU CAN AFFORD
THE WORLD FAMOUS
Diablo Printer Terminal



IPSI PROUDLY ANNOUNCES A PRICE BREAKTHROUGH ON DAISY WHEEL PRINTER TERMINALS

Featuring: **RS 232** compatibility at 110, 150, 300, and 1200 baud: **Print speed** of 45 char/sec: **Graphics** capability: Over 30 **print wheels** from 10 to 15 char/inch: 5 ribbons in color, cloth, film or two color: **Form length** adjustable: **Spacing** adjustable from 120 pitch to .1 pitch: Line spacing adjustable from 48 lines/in. to .4 line/in.: Identical in every respect to it's high priced sister the **Diablo HyTerm II*** model 1620.

Price: $2,595.00 each in single units, quantity discounts available.

Service: Major cities in the U.S. and Canada.

You have probably waited a long time to purchase your computer printer at a reasonable price. Wait no longer! The Diablo 1620 lists for $3,240.00 we sell the IPSI 1622 for $2,595.00; save well over $600.00. For immediate action phone in your order collect and we will ship your terminal the same day.

We also Stock Diablo 1345A (Hytype II*) mechanisms for immediate shipment at the low price of $1,150.00. Order one now and build your own terminal using your own CPU, or do-it-yourself by buying our 1622 kit and assemble your own terminal within 10 hours for $2,295.00

If you've always wanted to turn your computer onto word processing, this is the only way to go. Ask for our free word processing software.

IPSI

INTERNATIONAL PERIPHERAL SYSTEMS, INC.
1849 N. HELM
FRESNO, CALIFORNIA 93727
CALL (209) 252-3635

Circle 181 on inquiry card.

possibly reach the exit. Therefore, it doesn't matter that these were unexplored, and every intersection along the loop can be deleted from the optimized path with no loss of important information.

Detecting the two different types of backtracks (straight line and loop) involves answering the same question: has the mouse been here before? The handling of both types of backtracks is also the same too: delete all of the path history between the last visit here and this visit, then continue from there. Up until now, the optimizing process has been discussed in an "after the fact" fashion, as if the whole maze walk path had already been generated from the start to the finish. However, it can be much more efficient if the mouse can perform these optimizations while it is recording the path.

The actual details of the implementation are not important here, since they would depend on the type of microprocessor used. In general, though, assume that the intersection X,Y coordinates are stored in the mouse's memory in a linear fashion (ie: in an array or list). Furthermore, assume that there is a pointer into the array or list indicating where the next coordinate pair will be stored (ie: NEXT pointer). In this way, each time the mouse encounters an intersection and is about to record its coordinates as the NEXT position in the path list, it can scan backward from the NEXT pointer to the beginning of the list, looking for an occurrence of the same coordinates. If no occurrence is found, this is the mouse's first visit here. These coordinates can be recorded at the NEXT position in the list, and NEXT can be appropriately incremented to prepare for any successive intersections. If an occurrence of the same coordinate pair is found (ie: at position I in the list), the mouse has been here before. The easy way to "forget" the backtracked part of the path (between I and NEXT) is for the mouse to reset NEXT to I, then continue normally by incrementing NEXT and looking for another intersection. Figure 8 shows this diagramatically.

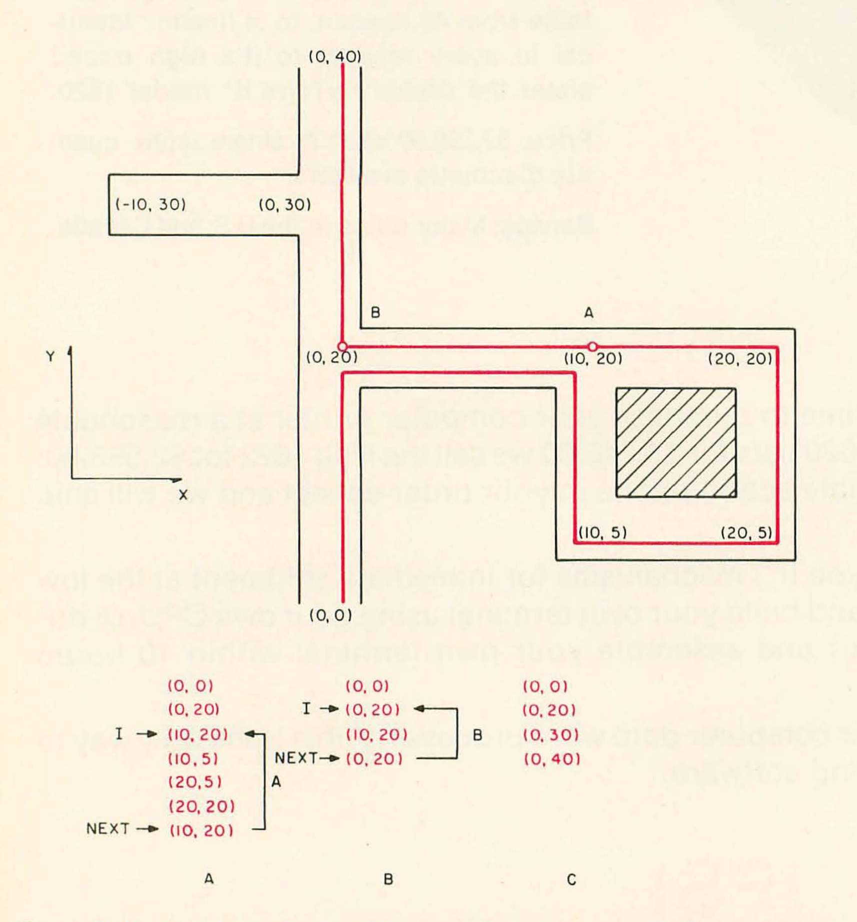


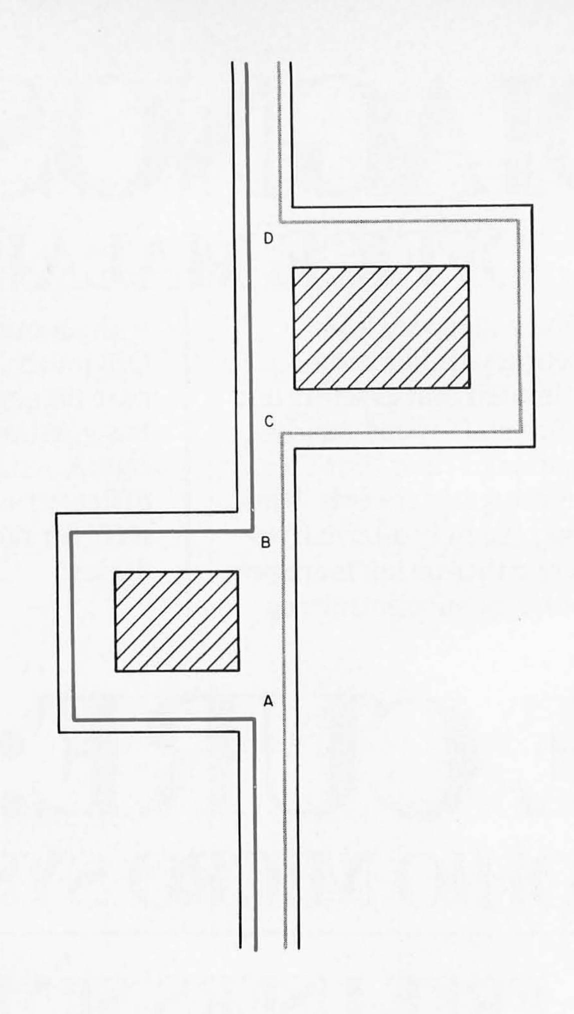
Figure 8: Backtracking checks. Whenever the mouse records the intersection it is in, it checks backward in the path list to see if it has been here before. If so, the path list is pared back to that point, and thus the backtracked path is automatically "forgotten." For example, in A, the mouse is at intersection (10,20); it scans back through its path history and sees that it has been here before at pointer I. So the NEXT pointer is reset to just after I and the mouse continues. In B, the mouse again finds that it has revisited an intersection, and again the NEXT pointer is reset before continuing. In C, the final, optimized path is shown.

Incidentally, if the mouse has relatively low accuracy motors and sensors, it is possible to obtain slightly different readings when encountering an intersection for the second time. Therefore, when checking to determine if this intersection has been visited before, allowances must be made for the inaccuracies. This is easily accomplished by checking to see if X[NEXT] is within plus or minus delta of X[I], and if Y[NEXT] is within plus or minus delta of Y[I], where the delta value reflects the amount of possible deviation, instead of checking for X[NEXT] = X[I] and Y[NEXT] = Y[I].

Although it may seem that a lot of computation is done while the mouse is running the maze (and, after all, speed counts), in fact, the time taken for computation is so small compared to the time it takes for the robot to move to the next intersection that it is hardly noticeable. Another factor which makes doing the optimization during the run even more desirable is that the exit of the maze is not always well defined. This means that the robot merely passes through a corridor and trips a light sensor to stop the timer, and then proceeds to "fall of the edge of the world." This makes it difficult for the mouse to determine that the maze run is finished, and that it should now optimize the recorded path. Optimizing during the run is certainly cleaner and more efficient.

A suitable way to take advantage of the *backtrack-trimming* algorithm would be

Figure 9: Optimization. If a leftmost path were to yield the red path and a rightmost path were to yield the gray path, the common intersections would be A, B, C, and D. By taking the shorter path between each successive pair of intersections, one obtains the shortest total path which has been traversed. Examples of short paths are the gray path from A to B, either path from B to C (since they are the same), and the red path from C to D. The final path is the most direct and shortest path through this section of the maze.



for the mouse to take the rightmost path, optimizing and recording as it went on the first attempt, and similarly take the leftmost path on the second try. Now the mouse has recorded in its memory two different optimized paths. At the beginning of the third attempt, each of the two paths can be measured by a straightforward, length of corridors sum, and the shorter of the two optimized paths can be taken on this final run.

A logical extension of this shorter total path philosophy is to compare the leftmost path list to the rightmost path list, finding all the common intersections. A short total path can then be composed by joining the shorter of the path segments between each common intersection pairs. For example, each complete path might go through the origin, intersection A, intersection B, and the exit point (perhaps along with many other different intersections). Then an optimal path could be made by combining the shorter path between the origin and A, the shorter path between A and B, and shorter path between B and the exit. Conceptually, this is like breaking the maze down into "common denominator" sections and picking the shortest path through each individual section. And, in fact, this path is the shortest one which has been traversed through the maze so far (see figure 9).

However, time is critical, not distance, and most likely the robot corners more slowly than it goes straight (see figure 10). So if the shorter path has many turns in it, and the longer run has few turns, the shorter path may not necessarily yield the shorter time for the run. Therefore, a slightly more sophisticated scheme could measure the paths using a weighted sum (a larger value for turns than for straightaways), and yield values which more closely reflect how fast the robot can negotiate the maze by the different paths.

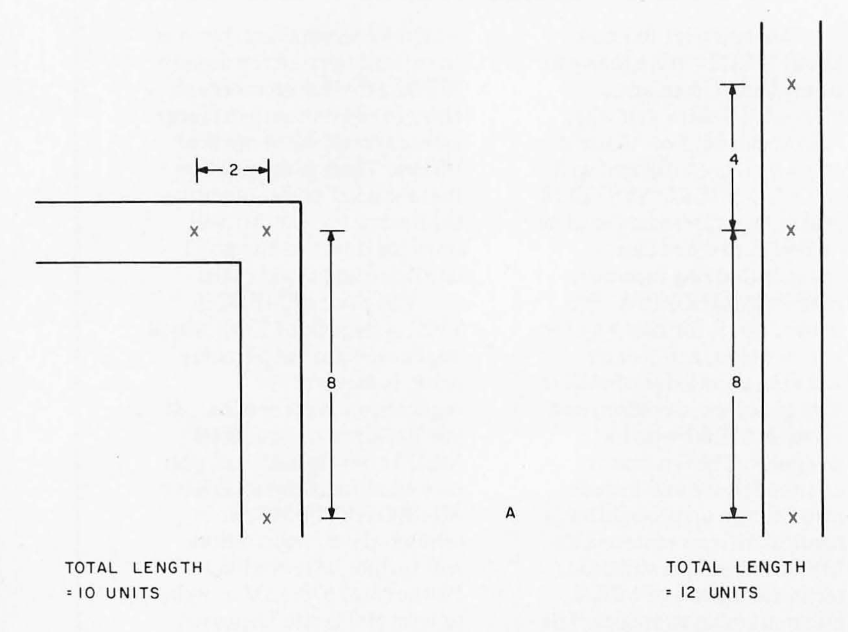


Figure 10: Turning time considerations. Due to the time it takes for the mouse to turn in an intersection, evaluations regarding this difference in the average length traveled per unit time must be made. For example, the mouse moves one unit of length in one unit of time and it takes two units of time to turn in an intersection. Then with the mouse starting in position A in each course, after 12 units of time the mouse has progressed 10 units of length on the left and a full 12 units on the right.

Conclusions

The algorithms presented here are by no means high-powered or devious, but are more the results of a natural, intuitive approach to the maze traversal problem. They are all straightforward and relatively easy to implement. But even so, they are reliable and produce solutions which are reasonably good, especially when compared with the common wall-hugging tactics.

Clearly, there are still many ways to improve the performance of these algorithms. More contests like the *IEEE Spectrum's* Micromouse contest will perhaps encourage investigation in this area and will produce much more sophisticated approaches and solutions to the maze problem.■

NorthStar

Gazer

"Micro-Computer Products to Look Up to"

Horizon Disk Capacity Keeps Growing

The Horizon is now capable of 720K bytes on-line! The Horizon can connect to four double density 5¼" single-sided disk drives. Each of those drives can access 180K bytes of information. A four drive system accesses 720K bytes!

That's capacity you don't usually find in a microcomputer, but there's even more to come! The North Star disk controller board is designed so that two-sided disk drives may be added as soon as they become available from North Star.

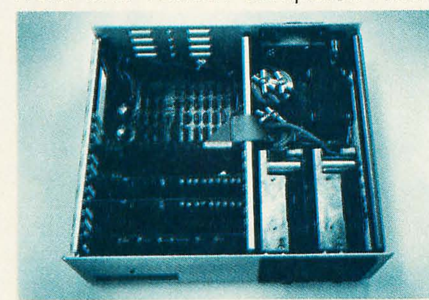
Existing Horizons will accommodate the new two-sided drives so North Star owners can simply add additional drives to up-grade their system. Each two-sided drive will access 360K bytes! That means the maximum on-line disk storage for the Horizon will increase to over 1.4 **million** bytes!

New Cabinet for Disk Drives

North Star additional disk drives are now available with the same high quality wood cover as the Horizon computer! The Additional Drive Cabinet (ADC) is designed to accept either one or two drives for the Horizon or for mounting North Star Micro Disk System drives. Like the Horizon, the ADC is available with either wood or blue metal cover. Included is a new power supply capable of powering one or two drives. The ADC is $129 in kit form. Assembled, with one drive the ADC is $599, with two drives $999.

Pascal Now Available for Horizon

The much-heralded Pascal language is now being offered for use with the North Star Horizon computer. North



Inside view of Horizon with processor board, RAM board, disk controller, two drives, and power supply.

Star, with the co-operation of the University of California at San Diego, is now delivering a Pascal Program Development system. North Star Pascal is ideally suited for developing large programs because of features such as: long variable names, block-structured control statements, and compilation. North Star Pascal is available on 5¼" diskettes for use with the Horizon or Micro Disk System. North Star Pascal will operate with either the Z80 or 8080 microprocessor.

Pascal, including documentation, is available in either single or double density versions for $49.

An auxiliary Pascal diskette, containing an 8080/Z80 assembler and some additional Pascal utilities, is available for $29. Complete information is available at your local retail computer store.

First Double Density, Now Double Memory

The new North Star 32K RAM board (RAM-32) has doubled the memory density of the popular Horizon computer. Available either with the Horizon or other S-100 bus computers, the RAM-32 runs at full speed — no wait states — with the 4 MHz Z80A microprocessor (as well as with slower Z80 and 8080 processors). Addressability of the RAM-32 is switch-selectable in four 8K regions.

North Star RAM features like bank-switching and parity checking are standard. The parity checking capability means that the RAM-32 is constantly diagnosing itself. That's a plus for your system. The fact that parity checking is a North Star RAM-32 standard is a plus for your pocketbook! There is no extra charge for this important capability.

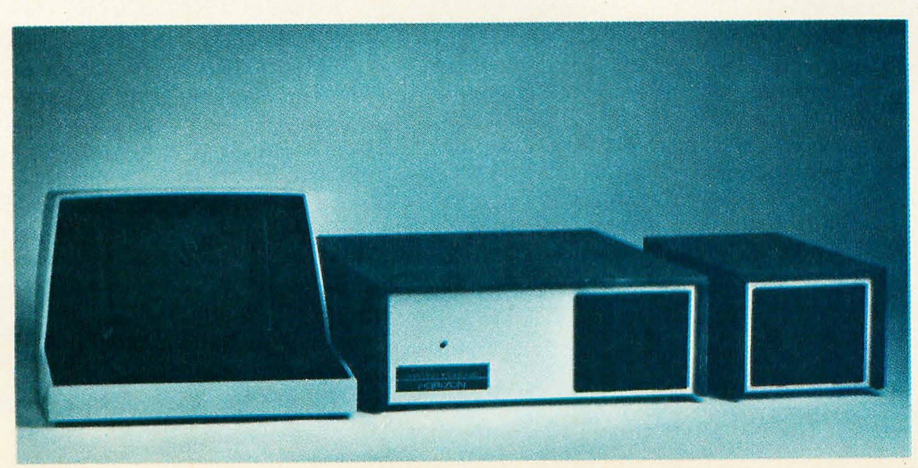
A Horizon with 48K of RAM can be configured by using one North Star 16K RAM board and a RAM-32. Need more memory? 56K can be configured by using two RAM-32 boards with one 8K region switched off.

NORTH STAR MDS, ZPB, FPB FOR OTHER S-100 COMPUTERS

Upgrade your system with these North Star products — available for any S-100 computer: Micro Disk System — a complete 5¼" floppy disk system, Z80 Processor Board, or the Hardware Floating Point Board.

Horizon and RAM board prices are:

	Kit	Assembled
Horizon - 1-16K	$1599	$1899
Horizon - 1-32K	1849	2099
Horizon - 2-32K	2249	2549
RAM-32	599	659
RAM-16	399	459



◀ A typical Horizon configuration: CRT, Horizon computer, Additional Drive Cabinet (ADC).



Computers

2547 Ninth Street
Berkeley, California 94710
(415) 549-0858

Circle 285 on inquiry card.



With A New Concept In Documentation

PRS MAKES A BREAK-THROUGH IN DOCUMENTATION

Every PRS program comes with a complete and instructive handbook. This unique documentation is written in clear and easy-to-understand English. PRS offers you a true fluency in computer-user dialogue.

PRS HAS A POWERFUL CODE

Powerful code means fast and accurate answers. PRS permeates the design of its unique code with "human engineering." And you'll appreciate the delicate balance between machine-human dialogue and resident program-space memory.

PRS PROGRAMS ARE PROTECTED IN A BEAUTIFUL CASSETTE OR DISK FOLDER

The PRS cassette or floppy disk and manual for your PRS programs are protectively nestled in an efficient and durable gold-imprinted folder. Wherever you file it, in your room or office, the decor will be enhanced by its elegance.

PRS SUPPORTS YOUR MAJOR BRANDS

Over 20 packages are currently available. For example: Cuts, Xitan, Tarbell, TRS-80, Micropolis, North*Star. Use the order form on the other page to order your program. Just indicate your version and the configuration of your computer. Mail it to us today. And we will have it fulfilled immediately by a dealer in your area.

VISIT YOUR NEAREST DEALER TODAY

Let your dealer demonstrate the excellence of PRS programs. Let him show you the new concept in PRS documentation. Use the order form to obtain the names of the dealers in your area.

PRS EXCITING LINE OF PROGRAMS

☐ MICROFILE: A NEW DATAFILE MANAGEMENT PROGRAM

With PRS MICROFILE you can: Manage any list of items. Edit files. Sort items alphanumerically. Search by keys. Total columns. Justify columns. Save and retrieve data with mass storage media.

PRS MICROFILE documentation also solves one of the major concerns of beginners. A special chapter explains how you can interface newly acquired software with your specific configuration.

PRS gives clear tutorial instructions on the whole matter of I/O patches. In addition, you will always find a complete source listing of the I/O section of our programs. PRS makes every effort to facilitate the implementation of its software within your particular system.

Cassette $49.95
Floppy Disk $64.95

☐ DDS II: THE FAMOUS DYNAMIC DEBUGGING SYSTEM

DDS II is incredibly powerful. Although designed for microcomputers, it is even more sophisticated than the built-in debugging environments supported by most large main frames.

DDS II assembles and disassembles all 8080 code. Performs "software open-heart surgery", artfully displaying on the screen all registers, program instructions and memory. Simulates dual screen capability. You can switch on command from the DDS II display to the program output. It's like having two video monitors for the price of one. Updates the display while monitoring the program. Includes a full array of monitoring tools.

Cassette $49.95
Floppy Disk $64.95

☐ CORE: AN INDISPENSABLE "BRAIN-SURGEON"

This PRS program is the TRS-80 System Monitor—an essential program for the TRS-80 level II.

It permits the user to penetrate and manipulate the contents of the memory. It can question the input and output ports, the CPU registers, control the mass storage drives and much more.

Here's a partial list of commands: Examine and modify memory. Display memory in Hex, ASCII, or CPU registers. Enter ASCII in memory. Fill memory. Move block of memory. Verify memory. Locate string. Branch to routine. Go to and execute (two breakpoints available). Hex arithmetic. Question input ports and command output ports. Tape Cue, on/off. Read file in memory (bias available). Write file from memory. Memory test.

Cassette $38.95
Floppy Disk $64.95

☐ A2FP: A PLOTTING PROGRAM FOR APPLE II

This PRS program is a "modern age" tool for students, engineers and researchers. A2FP plots 2-dimensional functions in HIRES graphics. Uses the full screen (280 x 192 points.) Plots on Cartesian coordinates. Gives complete parameters of plotted curves. Superimposes plots. Offers many plotting modes.

Cassette $34.95

☐ T/PM: A TRANSIENT PROGRAM MANAGER FOR THE TRS-80

T/PM is a disk operating system similar to those used by main frames. It is written to utilize the full power of the Z-80 instruction set.

At last, software can now be transportable. All programs written for the C/PM system can now be relocated just once to run on the TRS-80. If a program is written for the TRS-80 using T/PM or other CP/M* like DOS, you can now run that program simply, without patches.

Floppy Disk $149.95

*Trademark of Digital Research

Circle 308 on inquiry card.

PROGRAMS



PRS—THE PROGRAM OF THE MONTH CORPORATION
257 Central Park West, New York, N.Y. 10024

SOFTWARE	PRICE	QTY.	TOTAL
1. MICROFILE: A new Data File Management System			
☐ Cassette	$ 49.95		
☐ Floppy Disk	$ 64.95		
☐ Documentation only	$ 10.00		
2. DDS II The Famous Dynamic Debugging System			
☐ Cassette	$ 49.95		
☐ Floppy Disk	$ 64.95		
☐ Documentation only	$ 10.00		
3. CORE: The TRS-80 Co-Resident System Monitor			
☐ Cassette	$ 38.95		
☐ Floppy Disk	$ 64.95		
☐ Documentation only	$ 10.00		
4. A2FP The APPLE II Function Plotter			
☐ Cassette	$ 34.95		
☐ Documentation only	$ 10.00		
5. T/PM: The TRS-80 Transient Progam Manager			
☐ Floppy Disk	$149.95		
☐ Documentation only	$ 50.00		

☐ Send me list of dealers in my area
☐ Check enclosed
☐ VISA ☐ Master Charge ☐ U.P.S. COD

Account #__________ Exp. Date__________

Bank #__________

Sub Total	
Less 5% with order	
N.Y. State sales tax	
Total	

Signature __________

My Computer configuration (specifying disk system) __________

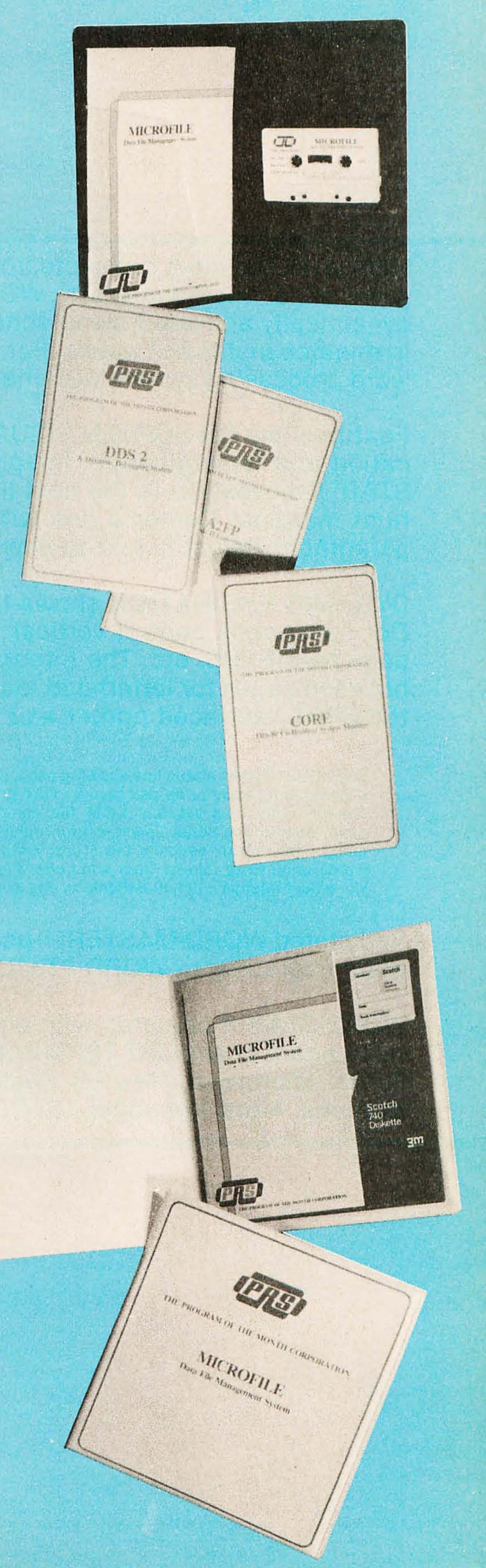
Name __________

Corporation __________ Telephone__________

Address (no P.O. Box) __________

City __________ State__________ Zip__________

689



Circle 308 on inquiry card.



MicroPro International Corporation

The Producers of WORD-MASTER T.M. and SUPER-SORT T.M.

Proudly Present

WORD-STAR T.M.

Here it is! The most complete, totally integrated, word processing system software you've ever seen on a microcomputer. WORD-STAR T.M. really shines in ease of use, with its unique, dynamically activated menu scheme that allows typists to become word processing operators in minutes instead of weeks. For only $495, your 8080/8085/Z-80 *with any CRT device* becomes a word processing system with the features of a Lanier, Wang, Vydec, Xerox, etc, for far less cost.

Features like ON-SCREEN VISUAL TEXT COMPOSITION, with word wrap, plus dynamic justification and remargination. Imagine being able to type without hitting RETURN, and as WORD-STAR T.M. moves you to the next line, the preceding line is redisplayed, justified to your left and right margins! Center a line with a keystroke! Set **BOLDFACE** and/or underline even in mid paragraph! Rejustify text to new margins, on-screen, at will!

DYNAMIC PAGINATION shows the printer page breaks during text entry, correction, or review. Dot commands control vertical page layout, CONDITIONAL PAGINATION, page HEADINGS, page NUMBERS, etc. The integrated PRINT COMMAND can print selected pages only, pause between pages for letterhead loading, and drive any CP/M* list device. WORD-STAR T.M. runs with the CP/M interfaced printer and terminal of your choice.

SPECIFICATIONS: status line showing page, line, column of cursor; variable menu suppression for experienced operators; dynamic display of text during entry or on call permits CRT preview of printed copy; full disk buffering (document sizes up to diskette capacity); multiple file/disk edits in a session; CP/M file compatible — also useful for data entry, editing programs, etc. Commands include: cursor up/down/left/right; scroll line/screenful, up/down, once/variable rate; insert/overtype; delete character/word/line left/right/entire; variable margin set/release; set/clear tab stops; justify/ragged right; block move/copy/delete; search/replace once/n times/global/selective/whole words only; write to/read from additional files; set/go to any of 10 place markers. Additional support provided for CRT's with inverse video/dual intensity, line insert/delete, and erase to end of line. Dialog interface for easy installation. AND MORE!

Registered WORD-MASTER T.M. users can get a $100 discount from us or your participating dealer when upgrading to WORD-STAR T.M. Offer good for a limited time only.

WORD-STAR T.M. operates with any CP/M* operating system, 32K, and either a memory mapped video board or any CRT TERMINAL with cursor control and clear screen (Hazeltine, Lear-Siegler, Soroc, Microterm, Hewlett Packard, Infoton, Beehive, Processor Tech VDM, IMSAI VIO, SD Systems, Altos, Dynabyte, GNAT, etc). We ship on 8" IBM, Micropolis II, or Northstar diskette.

Other fine MicroPro Products include:

	Software/Manual		Software/Manual
Word-Star T.M.	$495/25	Super-Sort T.M. I	$250/25
Word-Master T.M.	$150/25	Super-Sort T.M. II	$200/25
Tex-Writer T.M.	$ 75/15	Super-Sort T.M. III	$150/25

MICROPRO INTERNATIONAL CORPORATION
1299 4th Street, San Rafael, California 94901
Telephone (415) 457-8990 Telex 340388 Dealer/OEM inquiries invited

"PROFESSIONAL QUALITY SOFTWARE YOU CAN COUNT ON, NOW"

See Demonstrations at NCC Personal Computing Booth 42

*CP/M is a trademark of Digital Research Corp.

 Circle 228 on inquiry card.

The circuit described is intended only for experimentation with a unique form of input. It should not in any way be used for diagnosis of bodily functions.

Mind Over Matter

Add Biofeedback Input to Your Computer

Steve Ciarcia
POB 582
Glastonbury CT 06033

I wouldn't want you to get the wrong idea from photo 1. I haven't given up computers and taken up telling fortunes. Just consider the photo as a slightly dramatized introduction to a topic we've all heard of, but know so little about: biofeedback. In layman's terms, this simply means having the capability to monitor (in this case electronically) physiological processes.

There are a variety of devices on the market referred to as brain wave monitors. Brain waves are but one of the many sources of energy categorized under biofeedback. Their common relationship is that they are all electrical pulses which run through the body as a result of brain or muscle activity. Nerves and muscles within the body generate electricity by electrochemical action similar to that in a battery.

When we want to lift an arm, the brain sends an electrical pulse to the muscles in the arm. Proper magnitude and duration of the signal result in coordinated activity. The actual energy that is transmitted from the brain is very small: on the order of a few hundred microvolts at the most. The most familiar of these signals is the voltage generated by the pumping of the heart. A graph of this voltage versus time is called an electrocardiogram (abbreviated EKG or ECG). An EKG looks like a spiked waveform, with periodic response equivalent to a heartbeat. Many individual muscle contractions contribute to a frequency spectrum of 0.1 to 100 Hz, with an amplitude of about 5 mV.

Another group of signals are the voltages generated using large skeletal muscles like biceps and triceps. A recording of these voltages is called an electromyograph or EMG. Occurring only when the muscles contract, not periodically like the heart, the frequencies are very low, but the voltage is higher: about 5 to 10 mV. Because of their magnitude, these signals are the easiest to monitor.

The last important biomedical signal is composed of very low amplitude voltages within the brain itself. These are recorded by the EEG (electroencephalograph). They exhibit both periodic and pulse mode. The 50 μV signals occupy a band that is generally between 1 and 30 Hz. The signals are further subdivided into delta, theta, alpha, and beta waves. These classifications signify activity in defined frequency bands. Differences in activity seem to reflect particular personality tendencies.

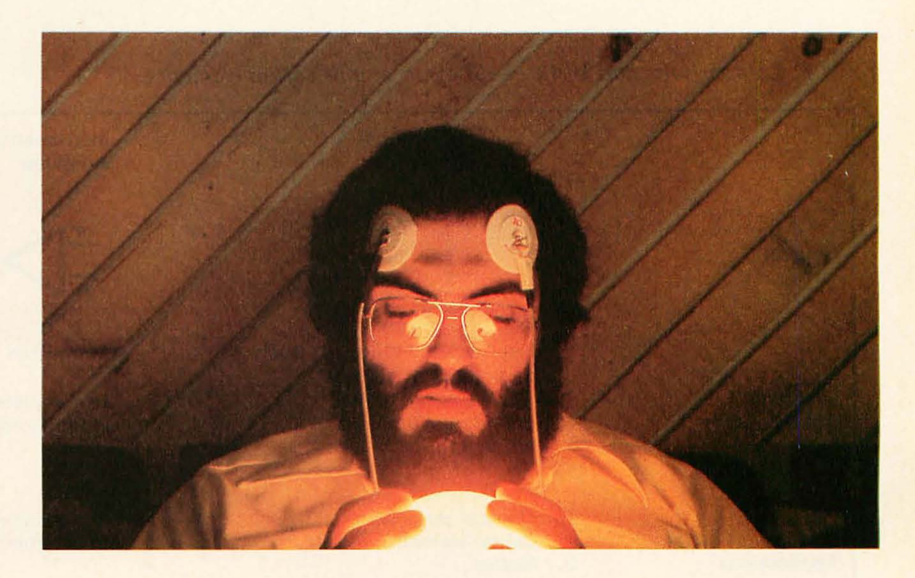


Photo 1: This photo simulates a crystal ball reflection to emphasize the control capabilities associated with this article.

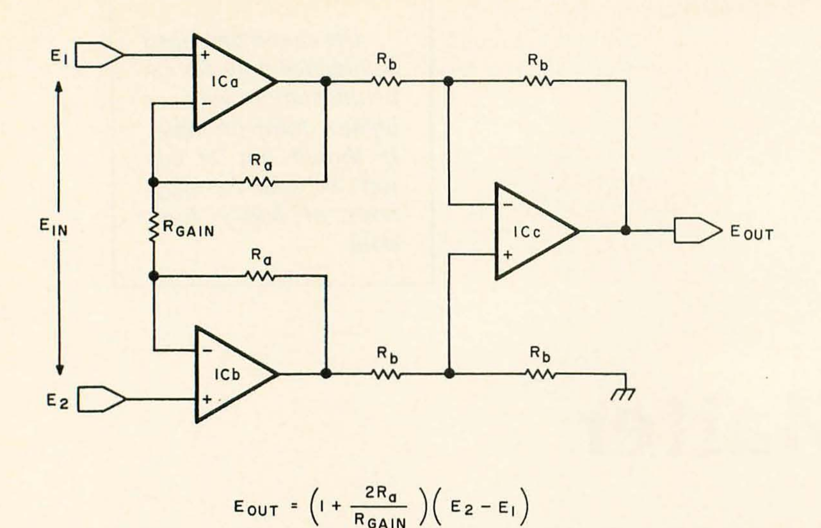


Figure 1: Differential input instrumentation amplifier configured from multiple, single-ended, operational, amplifier elements.

Monitoring Internal Electrical Activity

Consider the activity within the brain or the muscles. Each neuron is producing minute voltages. In combination with the voltages of billions of other cells involved in similar activities, the result is fairly significant. The situation can be compared to that of a football stadium before, during, and after a game. A listener outside of the stadium would not hear the shouts of a few individuals, but 50,000 people shouting is quite another story. A further consideration is the progress of the game. Loud noise coming from a particular section of the stadium during the game signifies approval. This same ovation, at the conclusion, can imply the identity of the winner. Observation and association are the keys. EKG, EMG, and EEG readings must be carefully interpreted.

All of the signals discussed thus far can be monitored with surface electrodes. When the biceps is moved, a small voltage which can be measured will be produced across it (ie: referenced to some other point on the body). Monitoring this voltage requires a special amplifier with extremely high input impedance and 60 Hz rejection. Care must be taken to use a device which will not load the signal being sensed, nor have such a low signal to noise ratio that one cannot discern intelligible information. The unique device which satisfies these requirements is called an instrumentation amplifier. Any product which is sold to monitor brain waves, EKGs, etc will contain an instrumentation amplifier.

Instrumentation amplifiers are often called differential or data amplifiers. They are closed loop gain blocks with accurately predictable input to output response. They are especially configured to have extremely high input impedances and common mode rejection which makes them ideal for amplifying low level signals in the presence of large common mode voltages. Figure 1 shows the schematic of a typical instrumentation amplifier built from such standard operational amplifiers as LM301s or 741s.

This common circuit consists of three op amps. ICa and ICb are inserted as high impedance input buffers which provide a differential gain of $1 + 2R1/R_{gain}$ and unity common mode gain. ICc is a unity gain differential amplifier which combines the voltages from the other amps. The ratio of the differential voltage gain of an amplifier to its common mode gain is enhanced

Table 1: Comparison chart of three different amplification elements.

	Operational Amplifier	Instrumentation Amplifier	Isolation Amplifier
Symbol			
Feedback Configuration	1. User defined feedback such as voltage or current. 2. Can be configured to provide dV/dt, ∫Vdt, log V, etc.	1. Committed feedback. 2. Gain adjustable within fixed limits.	1. Committed feedback. 2. Gain adjustable within fixed limits.
Basic Applications	1. General purpose amplification element. 2. Buffer. 3. Analog computational element.	1. High accuracy analog sense amplifier.	1. High accuracy analog sense amplifier. 2. Analog safety isolator. 3. Prevents ground loops.

by selecting low feedback resistors to reduce the effects of input offsets. A problem arises when selecting matched components to build this otherwise cheap circuit. Slight variations in resistors and op amps can make the difference between a working or nonworking circuit. (More on that subject will be discussed later.)

EEG and EMG monitoring requires an instrumentation amplifer because of the low input levels; but, when used in a biomedical application, a further modification to the amplifier's internal design is necessary. The special device is called an isolation amplifier. Transformers or optical couplers inside the amplifier block isolate the sense inputs of the amplifier from the output circuitry. This means that a 2 μV signal could be monitored on a 2000 V transmission line and the output connected directly to an analog to digital converter input on your computer. The protection works both ways. This is why any connections to the body are done through isolation amplifiers.

An isolation amplifier is to analog signals as an optoisolator is to digital signals. It prevents ground loops from the data analysis equipment (ie: your computer) through the subject. When the electrodes are attached, skin contact resistance is very low: only a few hundred ohms. A leakage current of just 100 μA can be fatal. Table 1 summarizes the differences between the amplifiers we've discussed.

Choosing an Isolation Amplifier

There have been many articles on the subject of alpha brain wave and muscle monitors; some even include circuit diagrams for construction of the interfaces. The major thing these articles lack is a caution about matching components, and the critical importance of proper layout. The circuit of figure 1, if breadboarded in the usual fashion, wouldn't have a chance of working on 50 μV levels. Even the testing of a handful of components to obtain matched pairs would be useless without concise wiring and plenty of ground plane shielding to reduce 60 Hz interference. Personally, I don't like to present circuits with so many strings attached that it takes divine intervention to make them work.

The final most important consideration in this undertaking is to not get electrocuted because of sloppy technique. At this point I'd like to draw the line between this article and other construction oriented articles. A cheap method of attaining minimal isolation is to use batteries to power an instrumentation amplifier. This sounds fine in theory, but it is very risky in practice. Too often a



Photo 2: The Analog Devices 284J isolation amplifier used in this article.

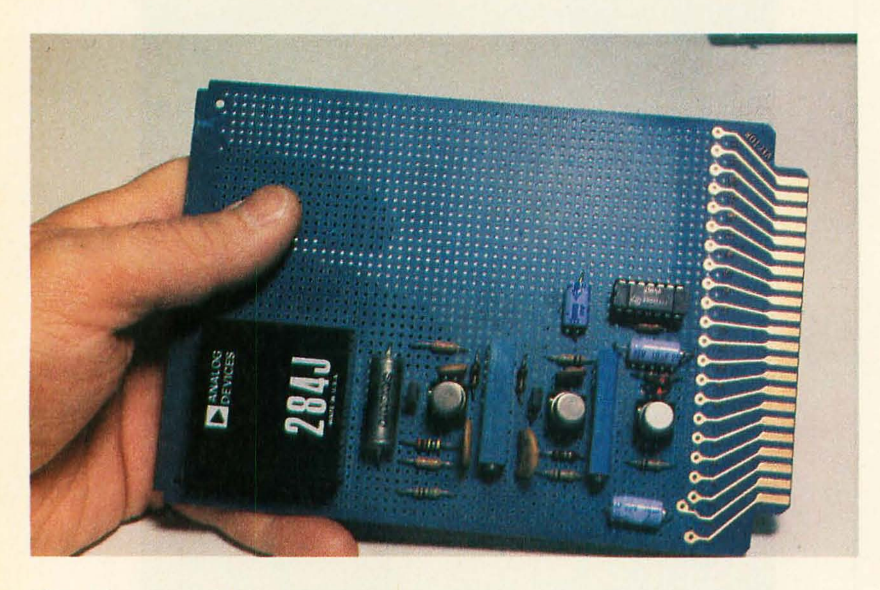


Photo 3: View of the prototype circuit described in figure 2.

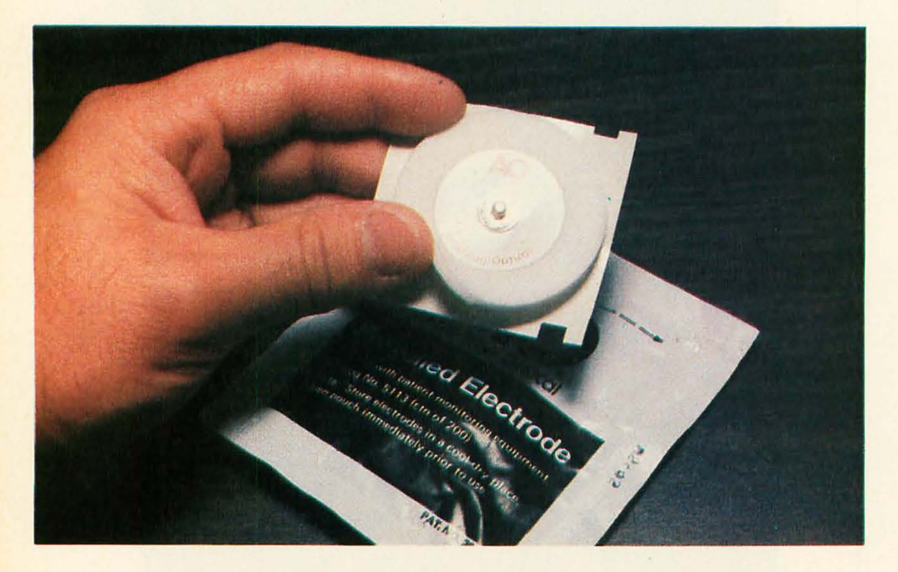


Photo 4: Pregelled American Optical electrodes of the type used in this article. They are available from medical supply outlets.

standard power supply is substituted for the batteries, or a loosely wired component falls against a live wire on another circuit.

Fortunately we can get both safety and performance if we don't assume that everything has to be constructed from scratch. It is a much better idea to take advantage of commercially available isolation amplifiers. (You wouldn't build a 4 bit digital counter from transistors, would you?) A perfect choice for this application is the Underwriters Laboratory approved Analog Devices 284J isolation amplifier shown in photo 2. It provides plus or minus 2500 V isolation, 110 dB common mode rejection, and a gain of 10 V per volt. For the experimenter this eliminates building the only tricky section of the interface. An added benefit is that the isolation is now an internal function of the 284J and not a function of installation. Since the ultimate aim of this article is to produce a biofeedback interface for a computer, I don't want anyone getting injured in the process.

Biofeedback Computer Interface

Figure 2 is the schematic of a circuit which is capable of sensing the minute voltages we've been discussing, and signifying to the computer when a present level has been attained. This is a bare bones, basic interface designed specifically for signal acquisition. It would seem to me that this is the area which would give most people problems. The circuit consists of an isolation amplifier module, two gain stages, and a comparator to sense peak level. The completed circuit is shown in photo 3.

All connections to the body are done through M1. The high and low input terminals are attached across the area to be monitored. If it is an EKG output, you should attach the terminals as shown. For biceps input, these two probes would go on the upper arm and the guard connected to the wrist. All leads between the body and the board must be shielded or 60 Hz will be all that is seen on the output. Gain on the 284J amplifier is set by connecting a resistor between pins 1 and 2. When they are shorted as shown, the result is a gain of 10.

ICs 1 and 2 are configured as common inverting amplifiers, each having a gain of 10. Since the signals we want to amplify are relatively low frequency AC, a capacitor is attached at the input of the first amplifier to filter out the DC component of M1's output. In most cases of muscle monitoring, this total gain of 1000 is sufficient. Picking up brain waves will require additional amplification. Changing the 100 kΩ resistor

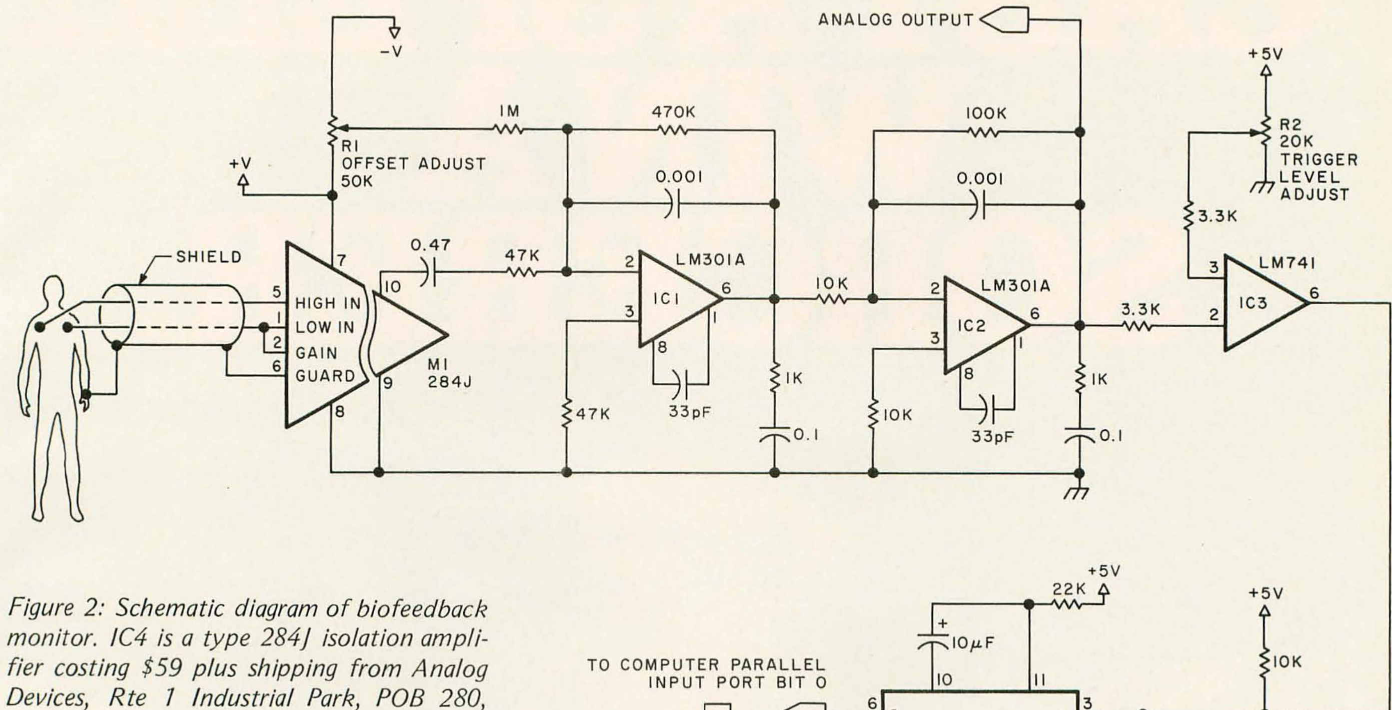


Figure 2: Schematic diagram of biofeedback monitor. IC4 is a type 284J isolation amplifier costing $59 plus shipping from Analog Devices, Rte 1 Industrial Park, POB 280, Norwood MA 02062. +V is anywhere from 9 to 15 V and −V is from −9 to −15 V. All capacitors are 100 V ceramic unless otherwise noted. All circuitry should be mounted on a ground plane to reduce AC pickup. Connecting wires should be as short as possible. The electrode cable must be shielded to obtain proper operation.

on IC2 to 1 MΩ will increase it another order of magnitude to 10,000. Be aware that raising the amplification also raises the noise on the output. Capacitors in the feedback loops are used in an attempt to keep this noise to a minimum. The amplified analog signal is available at pin 6 of IC2. It can be attached to an oscilloscope if you care to watch yourself in action.

IC3 and IC4 are the interface to the computer. IC3 is a comparator with normally high output. When the signal level from IC2 exceeds the trigger voltage set on R2, IC3 pin 6 goes low, firing the one shot IC4. This signal is in turn connected to a parallel input bit of the computer. Offset potentiometer R1 is adjusted to give 0 V on IC2 pin 6 when M1 is removed and M1 pin 10 is grounded.

Using the Muscle Monitor

Monitoring muscle voltages is much easier than monitoring brain waves. To adequately accomplish the latter, sharp bandpass filters which can separate brain waves from other signal sources must be added to

Number	Type	+5 V	GND	+V	-V
IC1	LM301A	–	–	7	4
IC2	LM301A	–	–	7	4
IC3	LM741	7	4	–	–
IC4	74121	14	7	–	–

Table 2: Power pin connections for figure 2 schematic.

figure 1. As it stands, it cannot differentiate between alpha or theta waves and is optimized for muscle pickup.

To sense the electrical activity of a muscle such as the biceps, three electrodes are necessary. It is not enough to merely wrap three wires around your arm. Special electrodes such as the type shown in photo 4 are necessary. These are referred to as pregelled silver-silver chloride disposable electrodes and they are available through medical supply outlets. The electrodes (shown in photo 5) have a spongy center section saturated with a gel to reduce skin contact resistance. The best results will be obtained by using these or similar attachments.

In the case of the forearm muscles, the

Text continued on page 56

CAST A SPELL. WIN A SORCERER.



If you've written software in Altair Basic, you've written "spells" for the Exidy Sorcerer.

Now, make it pay off!

There's never enough software. Particularly good software.

That's why Exidy is sponsoring a software contest where nobody loses.

Altair programs run on Sorcerer. The Sorcerer computer's Standard Basic is compatible with Altair 4K and 8K Basic. So our contest is open to programs—we like to think of them as "spells" or "Sorcery"—written in all three of those Basic versions.



WIN THIS EXIDY SORCERER.

Trade one of yours for one of ours. Just for entering a program in our contest, we'll send you a new, professionally written and documented program. Free. It's a classic game of concentration that's a fun mind-stretcher for both kids and adults. Plus you'll get our new 20" by 24" color poster.

And maybe 99 more good programs. We'll publish a bound book of the best programs entered—up to 100 of them, with full credit to each author. If you enter you can have a copy for just the printing and mailing cost. And if your program is included, you get the book free.

And maybe a free Exidy Sorcerer: Submit one of the four programs judged "best," and win a free Sorcerer computer. (Or choose Sorcerer accessories of equal value.) There'll be one winner in each of the following categories: Business, Education, Fun & Games, and Home/ Personal management.

Test-run your entry free. Take your program to any participating Sorcerer dealer if you want to give it a test run. At the same time, maybe you'll want to jazz up your program to take advantage of Sorcerer's state-of-the-art features. These include 512 by 240 high-resolution graphics; user-defined characters; and dual cassette I/O, among others.

You can turn in your entry right at the dealer's. And collect your poster and new program on the spot.

Enter now. Send us your entry with the coupon. Or visit your dealer. But cast your best spell at Exidy now. And see if you can't make a free computer appear on your doorstep.

RULES:

1) Entries, including documentation, must be printed by computer or typed double spaced on 8½ by 11 paper, with your name on every page.
2) Enter as many times as you like. This coupon, or a copy of it, must be completed and attached to all entries.
3) Enter at any participating Exidy Sorcerer dealer, or mail entries postpaid to the address on this coupon.
4) Entries must be received by midnight, Aug. 31, 1979. Winners will be notified by Nov. 30, 1979. For a list of winners, send a self-addressed, stamped envelope marked "Winners List" to the coupon address.
5) You warrant, by your signature on this coupon, that all program and documentation material included in your entry is entirely your own original creation, and that no rights to it have been given or sold to any other party, and you agree to allow Exidy to use, publish, distribute, modify, and edit it as it sees fit.
6) All entries become the property of Exidy, Inc. No entries will be returned, nor any questions answered regarding individual entries. No royalties, payments or consideration beyond the items set forth in this advertisement will be given to any entrant.
7) Judging will be by a panel of experts chosen by, and including representatives of, Exidy, Inc. Judges may assign programs to whichever entry category they consider appropriate. Decision of the Judges is final.
8) Employees of Exidy, Inc., its dealers, distributors, advertising agencies and media not eligible. Void where prohibited, taxed or restricted by law.

Circle 133 on inquiry card.



EXIDY, INC.
969 W. Maude Ave.
Sunnyvale, CA 94086

Gentlemen:
Here's my "spell." Send me my free program and poster. If I win, send my Exidy Sorcerer computer to:

NAME ________

ADDRESS ________

CITY ________ STATE ________ ZIP ________

DAYTIME PHONE ________

TITLE OF PROGRAM ________

CATEGORY ☐ Business ☐ Fun & Games ☐ Education ☐ Home/Personal Management

SIGNATURE ________ DATE ________

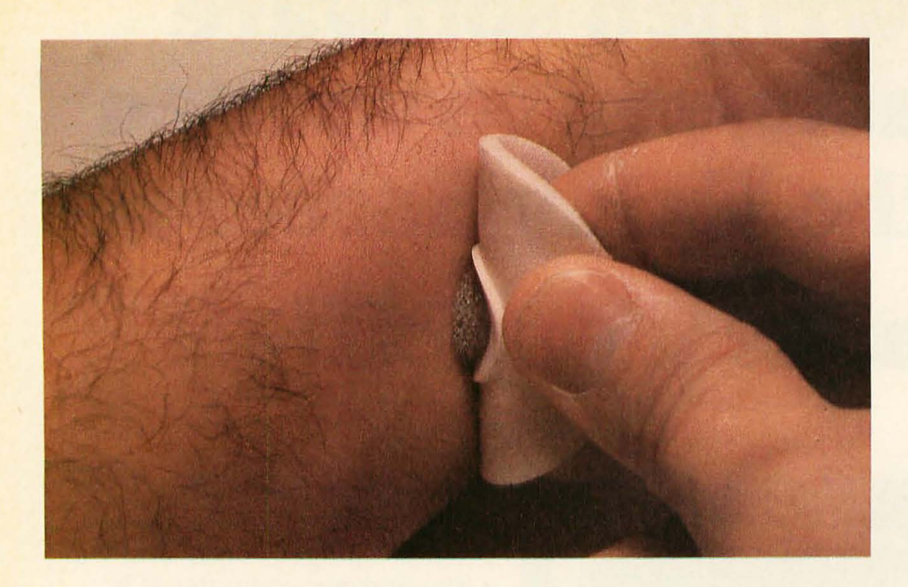


Photo 5: The electrode has a saturated spongy center which serves to reduce skin contact resistance. It is necessary to use this type of connection to the body if satisfactory results are to be obtained.

Text continued from page 53:

high electrode (shown in photo 6) is placed on the wrist, the low electrode on the upper arm, and the guard on my chest, close to the shoulder. When the muscles of that arm are flexed, a large pulse will appear at the analog output terminal of the interface. It is best seen with an oscilloscope. Every movement produces some noticeable deviation in the trace. If the trigger adjustment R2 is set above the ambient noise at the peak of this large pulse, it will fire the one shot every time the muscle is flexed. Actually, adjustment can be much finer. With the electrodes placed as in figure 1 (the guard is on my chest again), they can pick up something as insignificant as moving your eyebrows or gritting your teeth. The setting is made higher than the level produced when talking or breathing, so that it can be used as a suitable control input to the computer.

Biofeedback Computer Control

Control is the name of the game. Consider someone who is almost totally paralyzed. This system could be used (perhaps by sensing eyebrow movement) as an on/off switch to a more sophisticated controller. I've seen one computer aid for the handicapped which consisted of an alphanumeric sequencing display. Letters could be individually chosen and eventually combined to produce whole written messages. A lot can be accomplished with a single bit of input if the software is written with time as a pertinent consideration. A single switch could signify a particular choice if each was presented in sequence with time allotted to answer. That is the premise of the BASIC program in listing 1.

This is a simple program written in Micro Com 8 K Zapple BASIC. It presents the operator with a series of seven choices, and branches to special subroutines as a result of these choices. It presumes that the user can see and signify positive response by a high logic level on bit 0 of input port 3. This bit is tied to the output of our eyebrow twitch monitor. Output port 17 has seven lights attached to bits 1 through 7 (bit 0 not used). The program lights the first light, and the user decides whether or not the computer should perform the activity signified by bit 1. If so, the user merely furrows his or her brow and the program jumps to the designated activity. In this simple illustration, I merely flash the light a few times to indicate which was chosen. Should the operator not care for the first choice, the program sequences to the next choice, and so on. Before hook-

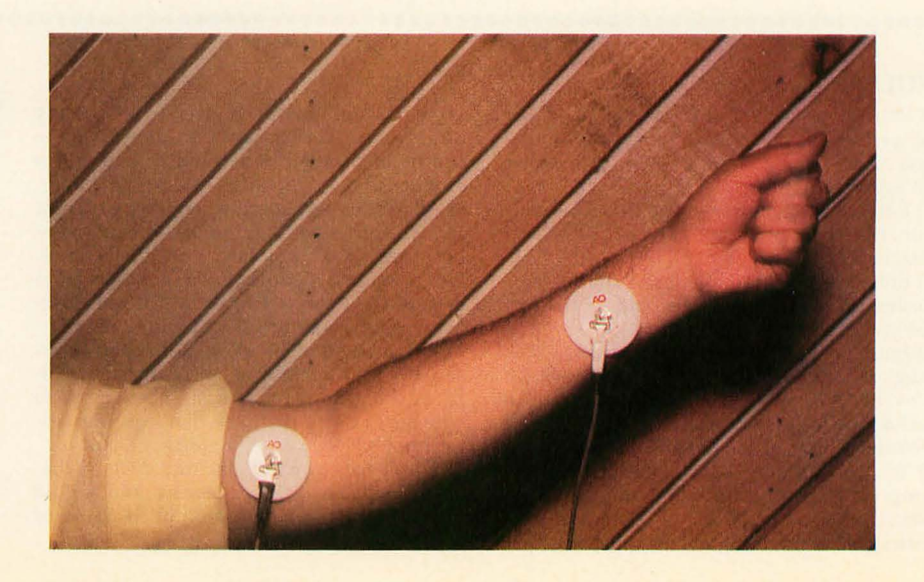
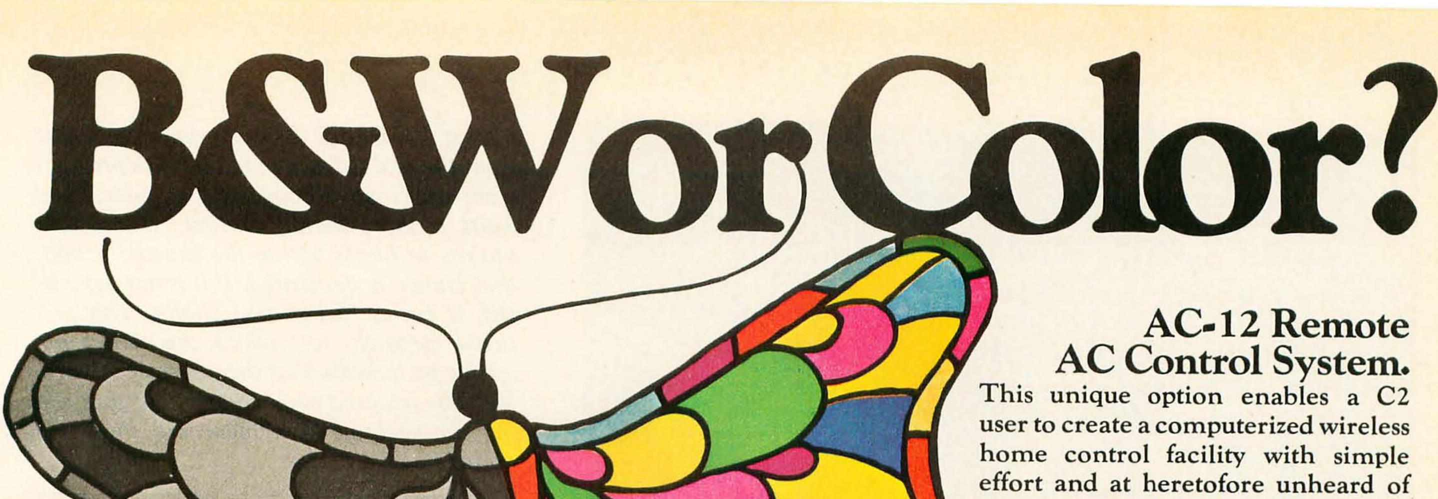


Photo 6: To monitor the electrical activity of the muscles in the arm, electrodes should be placed as shown.

B&W or Color?



AC-12 Remote AC Control System.

This unique option enables a C2 user to create a computerized wireless home control facility with simple effort and at heretofore unheard of low cost. The AC-12 provides a control signal on the home's AC power line not unlike the manner in which wireless intercoms function. The AC-12 will plug into any C2 computer that has the new 540B video board.

The basic system contains a command console and four remote modules. The remotes can be appliance modules that can switch up to 15 amps, lamp modules that can provide on-off, dimming and brightening, or wall switch modules. The AC-12 has a special Home Control OS-65D V3.0 Mini-Floppy Disk Operating System that provides a wide range of capabilities not the least of which is "Foreground and Background" operation. This allows the computer to monitor time and inputs for the Home Control System, while running other BASIC programs!

One of three exciting new options from Ohio Scientific.

Color. With it there is a big difference. And now, Ohio Scientific offers a lustrous 16 color option on all its C2 Series Computer Systems, past and present. New orders for C2-4P and C2-8P's can now request the color option. Upgrade kits are available for all existing C2's. When coupled with the C2's demonstrated high resolution graphics, the new color display is striking.

The relatively easy add-on of color and other options discussed here are interfaced through the new 540B video board and are realized without extensive modification or excessive cost because the C2 family of computers feature exceptional "design for the future" modular construction that permits one sector of the computer to be modified without affecting other sectors.

Sound. The sound option further enhances enjoyment of the C2's while broadening their scope of applications. Games of logic and dexterity now seem to come alive when sound pops out from on-screen happenings.

This new option is implemented through the new 542 Rev B keyboard available with new C2 orders or as part of an upgrade kit. In addition to a standard tone generator, this new board also contains a D/A (digital to analog) converter that takes advantage of the power and ultra high speed of the C2's 6502 microprocessor to enable users to generate both limited band-width speech and complex musical chords.

New options:

Color and Sound Option on C2-4P and C2-8P . . . . Add $200

CA-11B 540B color video board for system upgrades (does not include 542B sound keyboard) $225

CA-11C 540B color video board and 542B sound keyboard. (For upgrades) $299

AC-12 Remote AC control system, including console, four remotes and software. For use on any system with a 540B video board $175

Color. Sound. AC Remote Control. Three more examples of Ohio Scientific's continuing enhancement of its computer products and user benefits.

Contact your local Ohio Scientific dealer.

OHIO SCIENTIFIC
1333 S. Chillicothe Road • Aurora, Ohio 44202
(216) 562-3101

Circle 290 on inquiry card.

```
100 REM This program demonstrates how the computer can be
110 REM used to provide contol output from an EMG digital input
120 REM EMG input is on port 3, bit 0.  No stimulus is logic 0
130 REM while muscle activity is signified by logic 1.
140 REM Test apparatus uses 7 lights attached to bits 1 thru 7 of
150 REM output port 17. The computer sequences thru the lights until the
160 REM operator signifies a choice by --"THINKING"-- about it !!!
170 REM
180 REM Copyright 1979      STEVE CIARCIA
190 REM
200 REM
210 FOR D=0 TO 300 :NEXT D
220 REM
230 REM This routine sequentially flashes bits  1 through 7 of port 17
240 REM It only exits when an input flag has been set by the EMG monitor
250 B=1
260 X=2^B :OUT 17,X
270 GOSUB 440
280 IF F=1 THEN OUT 17,1 :GOTO 320
290 B=B+1 :IF B>7 THEN GOTO 210
300 GOTO 260
310 REM
320 IF B=1 THEN GOSUB 670 :GOTO 570
330 IF B=2 THEN GOSUB 670 :GOTO 580
340 IF B=3 THEN GOSUB 670 :GOTO 590
350 IF B=4 THEN GOSUB 670 :GOTO 600
360 IF B=5 THEN GOSUB 670 :GOTO 610
370 IF B=6 THEN GOSUB 670 :GOTO 620
380 IF B=7 THEN GOSUB 670 :GOTO 630
390 IF B>7 THEN STOP
400 REM
410 REM
420 REM This routine reads the EMG monitor on port 3 bit 0
430 REM If signal is present it sets flag F=1
440 A=0 :F=0
450 I=INP(3)-254
460 IF I>0 THEN 490
470 A=A+1 :IF A>200 THEN RETURN :REM give operator time to respond
480 GOTO 450
490 F=1
500 Q=INP(3)
510 IF Q>254 THEN 500
520 RETURN
530 REM
540 REM
550 REM These 7 routines can be replaced with outputs to
560 REM individual control programs.
570 PRINT"b=1":GOTO 210
580 PRINT"b=2":GOTO 210
590 PRINT"b=3":GOTO 210
600 PRINT"b=4":GOTO 210
610 PRINT"b=5":GOTO 210
620 PRINT"b=6":GOTO 210
630 PRINT"b=7":GOTO 210
640 REM
650 REM
660 REM This routine flashes individual light to indicate selection
670 FOR T=0 TO 10
680 OUT 17,X
690 FOR T1=0 TO 50 :NEXT T1
700 OUT 17,0 :
710 FOR T1=0 TO 50 :NEXT T1
720 NEXT T
730 RETURN
```

Listing 1: BASIC program to sense input from the biofeedback monitor. This program scans the cursor through several choices and waits a short period of time. If the user squints or blinks within the allotted period, that choice is designated. If it is not designated, it cycles to the next choice. This particular program just blinks the chosen objective to indicate that the interface is working. The required body connections for picking up eyebrow movement are shown in photo 1.

up, the program can be easily tested with the muscle monitor by temporarily attaching a normally closed, pushbutton switch on port 3 bit 0.

Conclusion

All of this effort for a single bit of data acquisition may appear unjustified, but it can prove to be exceedingly significant in situations where no other means of computer interaction is available. At the least, the interface should provide a substantial base for biofeedback experiments. With additional amplification and filtering to monitor brain waves, a whole series of challenging experiments come to mind. Personal computing need not be relegated to the level of canned amusements and commercial presentations. A refinement of this interface could be the one critical design feature which would open the field of personal computing to individuals who are otherwise physically unable to take advantage of it.

If you have any questions on this or any other "Ciarcia's Circuit Cellar" article, or just a good idea, please don't hesitate to write. While it may take some time, I do eventually answer all inquiries. Please enclose a self-addressed stamped envelope. Next month the "Circuit Cellar" topic will be sound generators. ■

THE $798* INTERTUBE II®

It's already a big success!

The party's over for all dumb terminals and a lot of smart ones too. But, at $798 (quantity 25), the party's just beginning for Intertec's InterTube II.

Standard features to celebrate include a full 24 line by 80 character display, 128 upper and lower case ASCII characters, reverse video, complete cursor addressing and control, an 18 key numeric keypad, special function keys, blinking, protected fields, character and line insert/delete, editing, eleven special graphics symbols, a 25th status line which displays the terminal operating mode and an RS-232 printer port.

You'll discover even more reasons to celebrate when you sit down in front of an InterTube II. Our wide bandwidth monitor produces crisp, sharp chararters everywhere on the screen. InterTube's Z-80 processor enables a host of operator oriented features to boost the efficiency of both software and programmers. And, InterTube's rugged modular design combined with its built-in self-test mode insures quick and reliable servicing.

InterTube's price/performance ratio can satisfy your requirements whether they be a sophisticated data entry application or a simple inquiry/response environment. So, there's really no reason to think "dumb" when you can afford to be so smart!

Join the thousands of InterTube celebrations going on around the country at this very moment. Call us at the number below and start your own celebration (BYOB—we'll bring the InterTube).



2300 Broad River Road, Columbia, S. C. 29210
(803) 798-9100 TWX: 810-666-2115

* $995 Quantity One

Circle 182 on inquiry card.

More Colors for Your Apple

Allen Watson III
430 Lakeview Way
Redwood City CA 94062

The sales literature for the Apple II lists the specifications for the high resolution color graphics mode this way:

- 280 horizontal by 192 vertical resolution.
- Four colors: black, white, violet, and green.
- Displays 8 K bytes.

The specifications don't sound all that exciting. The resolution seems about right, but why are there so few colors? And why did they pick green and violet instead of, say, red and blue? Well, as it happens, the colors in the Apple II high resolution graphics can be red and green, or blue and yellow, or almost any two complementary colors you want. What's more, on many color television sets you can obtain as many as four colors along with black and white, as demonstrated by the accompanying photograph.

The classic approach to computer generated color is to generate separate signals for the red, green, and blue inputs of a color monitor. However, color monitors are expensive; it's more economical to use an ordinary color television set. Now instead of generating three simultaneous video signals, we have to generate a composite signal that resembles the standard broadcast signal the television set was designed to receive.

It's not merely that the signal has to be put onto a regular television channel by means of a radio frequency modulator; although that's certainly necessary, there's a lot more to it. Since all the fascinating features of the Apple II high resolution color graphics are the results of the way the Apple II designers solved this problem, let's take a look at just what they did.

The Color Signal

The standards for broadcast color television signals were established by NTSC (National Television Systems Committee) and approved by the Federal Communications Commission in 1953. In order to retain the existing system of black and white television broadcasting, the committee system adds color information to a signal which is practically identical to the black and white standard. The resulting composite signal includes a black and white component that amplitude modulates the television carrier frequency in the usual way, and a color component which rides on a 3.58 MHz subcarrier.

This superposition of color and black and white information is necessary in order to crowd a full color video signal into a channel whose high frequency response is limited to just over 4 MHz. The fact that human vision does not resolve image details in color allows us to limit the resolution of the color component of the signal to a maximum of 1.5 MHz. In fact, only part of the color signal gets even this much; the rest is limited to 0.5 MHz.

This narrow band color signal modulates a 3.58 MHz subcarrier which is then added to the black and white picture information. The color subcarrier modulation is a combination of amplitude and phase modulation: the amplitude of the subcarrier corresponds to the amount of color at each point on the screen, while the choice of color is determined by the phase of the color frequency relative to a 3.58 MHz reference signal. This reference signal is generated in the television set from a burst of 3.58 MHz transmitted in the interval between the lines of the picture.

A high subcarrier frequency reduces interference between the color and black and white components because the black and white signal contains less energy at high frequencies. Interference is further reduced by the fact that the subcarrier frequency is an odd multiple of half the picture scanning rates, both horizontal and vertical. This makes any color signal that gets into the black and white video reverse polarity on successive lines; the interference makes little dots in the picture, but the dots on one line will have "undots" above and below. These will tend to average out when viewed from a reasonable distance.

This is where the signal generated by the Apple II deviates radically from the standard signal. First of all, the Apple II signal omits a technique called interlacing, thus reducing the number of horizontal scanning lines by half and likewise the amount of information needed to fill the screen. Noninterlacing is common among low cost computer video displays. The significant deviation from the standard, however, is a slight change in the horizontal and vertical scanning rates such that the interference between the color and the black and white components is maximized, rather than minimized. This is not as strange as it sounds, because this is what en-

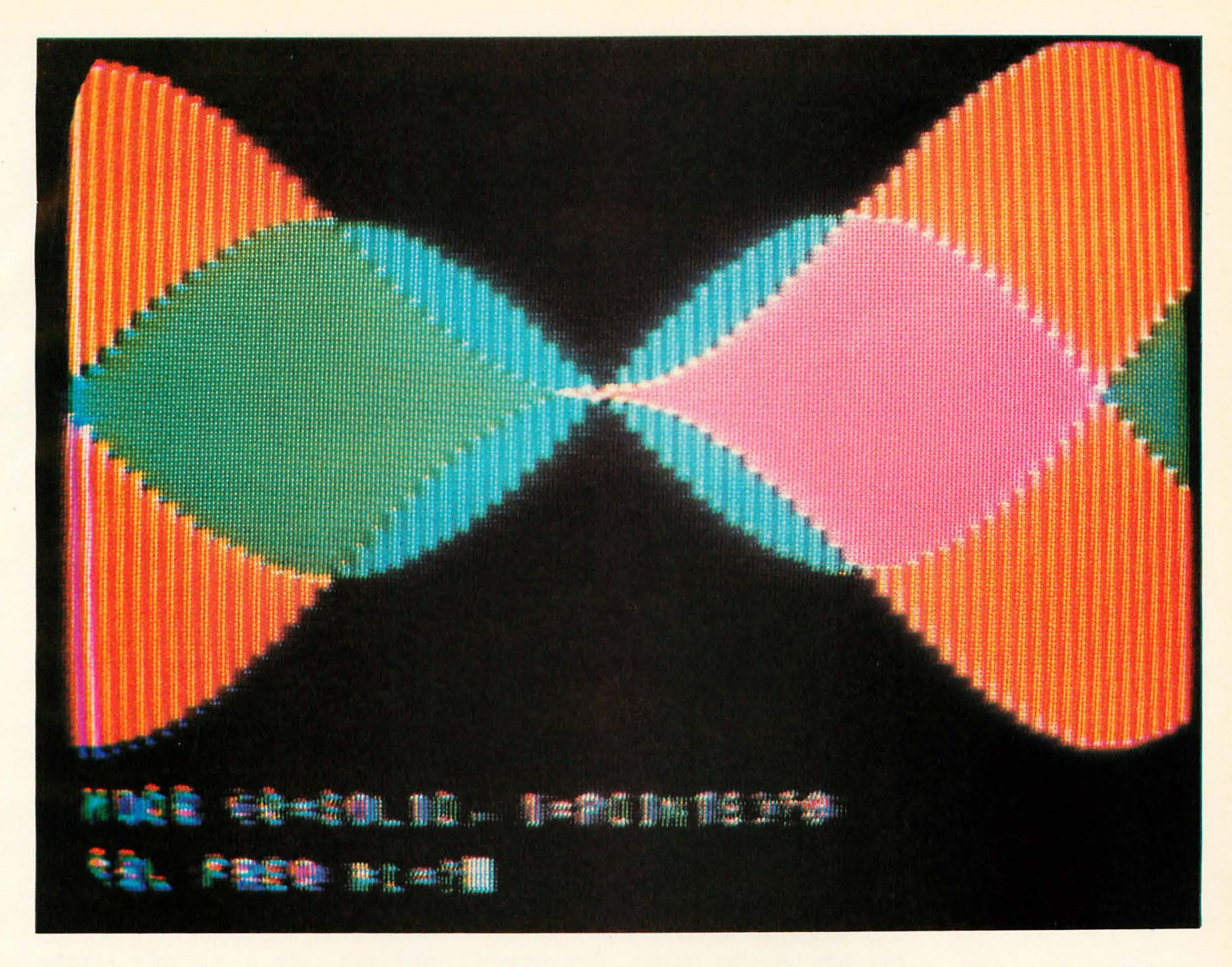


Photo 1: Apple II display showing four colors in high resolution mode.

ables the Apple II to generate color graphics with a signal made up only of ones and zeroes.

An Example

To see what this does to our display, suppose we try to display two small white dots side by side. The smaller the dots and the closer they are to each other, the higher the highest frequency picture signal going into the television set. But everything the Apple II puts out at the high end of the frequency range gets decoded as color, so that, even before our dots are made too small and too close together for a black and white set to be able to distinguish, something else has happened: they have merged into a single dot, and it isn't white, but color.

In other words, the resolution we can get using this method is somewhat limited compared with the separate red, green, blue approach, although it is produced with correspondingly less screen memory (ie: 8 K bytes compared with 24 K bytes). But even if the color interference were minimized, the television set's receiving circuits limit the horizontal resolution to about 300. Incidentally, this is why the Apple II displays only 40 characters in each line of text; the more popular 64 or 80 characters cannot be resolved by a standard color television.

Bits and Resolution

As we have seen, the Apple II produces color by simply putting its smallest dots at the right size and spacing: namely, the color subcarrier frequency. Each dot is really a half cycle, so the dot rate is twice the subcarrier frequency, or something over 7 MHz. Let's see how many of these dots will fit on one horizontal line. There is one horizontal scan every 63.5 μs, but part of this time is needed to get the electron beam into position to start the next line, and to keep the lines in synchronization. The picture signal is shut off, or blanked, during this time. That leaves about 45 μs, but just to play it

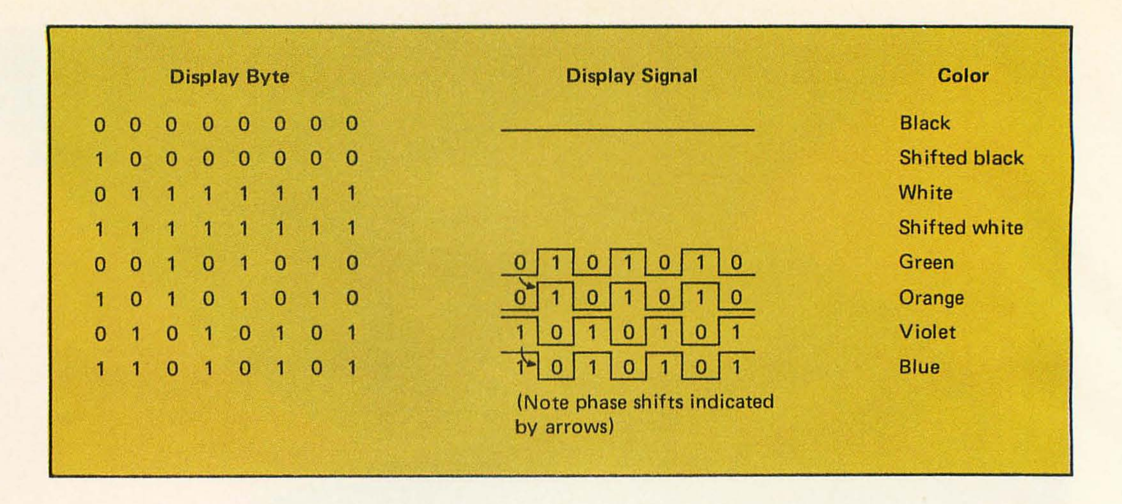


Figure 1: Colors produced by various bit patterns in relation to the color reference signal.

safe and to make sure that none of our valuable data gets cut off by the television set's normal overscan (the picture is set up to be bigger than the actual screen so there won't be any unsightly black borders), Apple II uses only about 40 μs of each line for data. This works out to 280 dots per line. In text mode, with 40 characters per line, this gives a character time of about 1 μs, which corresponds to the Apple II's system clock. Each character takes seven dot times, five for the character and two for spacing between characters.

The question is, exactly what does horizontal resolution of 280 refer to? Well, we can put a single dot at any of 280 different positions across the screen, but our dot will be colored, since it is a half cycle at the color subcarrier frequency. And if we put two dots too close together, they merge. Obviously, if the two dots are actually touching, no set could resolve them – this is really a single spot which happens to be two dots wide. But even if we put a black dot in between, we'll see only one dot, in color, because the dot spacing matches the color subcarrier frequency. Only by putting two or more black dots between our white dots will we be able to see a clear separation.

These relationships are diagramed in figure 1. The color reference signal is shown at the top. Any signal component at this frequency, even a single dot, will be displayed as colored. Theoretically, a double width dot contains no color frequency component, and hence will be displayed as white.

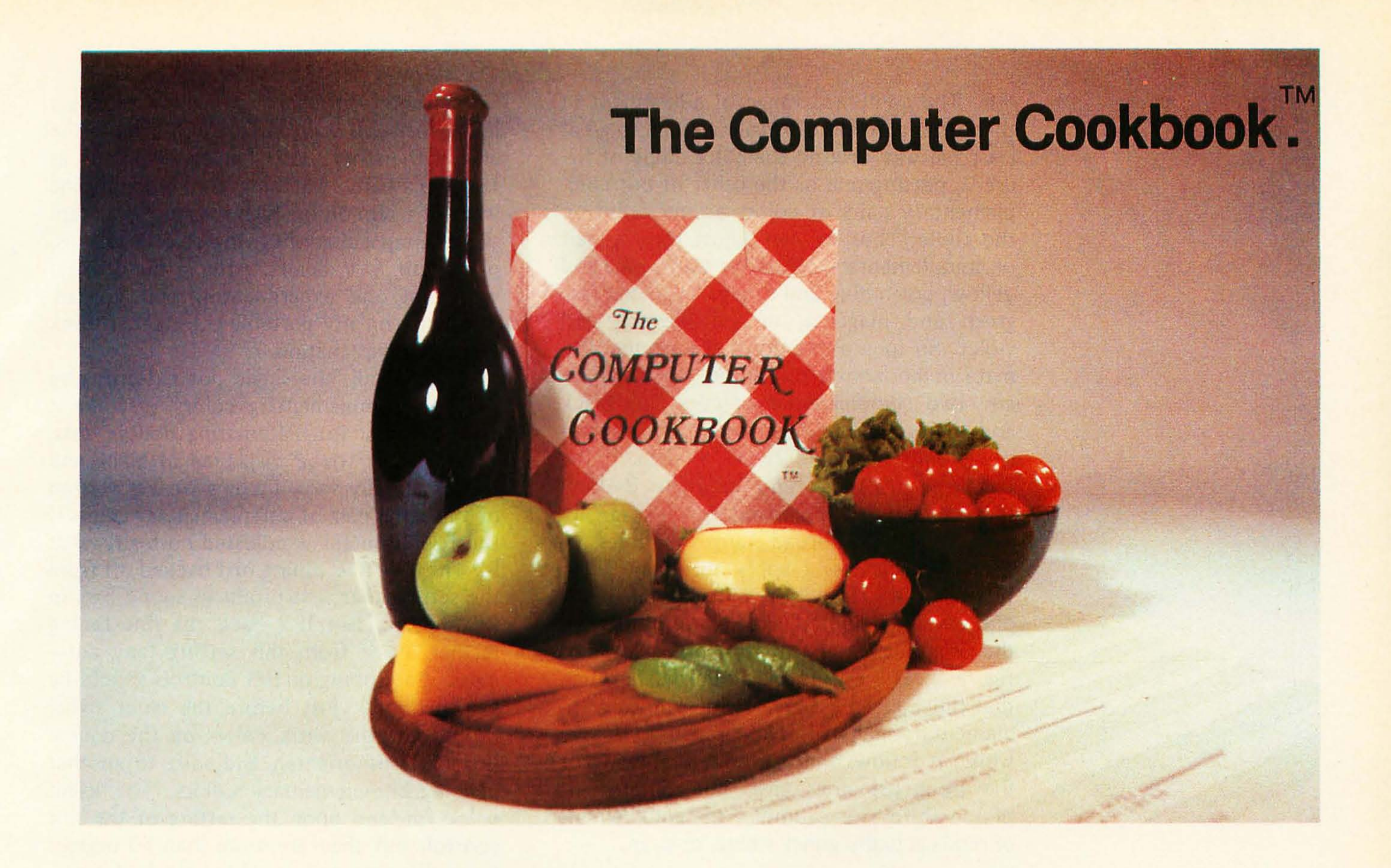
Apple II High Resolution Colors

Now we can see how the trade-off between color and resolution affects the way our computer bits are displayed by the television set. But let's look on the bright side: with the right bit patterns, we can put colors onto the screen. Let's ignore the resolution problem for a while and investigate the colors.

If we fill the screen memory with ones, the display will be all white; all zeroes paints it black. If we alternate ones and zeroes horizontally, we have a signal which is right at the color frequency, so it is displayed as a solid color. Now comes the interesting part – what color is it? As I mentioned earlier, the color is determined by the phase of the picture signal's color frequency component relative to the color reference signal, which is generated by the television set from the 3.58 MHz color burst which we transmit during the horizontal blanking interval. So our question becomes, "How can we control the relative phase of these two signals?"

First of all, our computer bits are output every half cycle of the color reference frequency. This means we can change the phase by 180 degrees by simply inverting the bit pattern so that alternating ones and zeroes become alternating zeroes and ones. Interestingly enough, since the color spectrum is allocated the 360 degrees of possible phase angles that we can have, complementing the bits also complements the color; that is, phase inversion amounts to 180 degrees of phase shift, and complementary colors are 180 degrees apart. The relation of color to phase angle is shown in figure 2. If the alternating bits are in phase with the color reference signal, the color will be yellow-green; out of phase bits will give us blue-violet. This determines the two colors Apple II specifies in addition to black and white. But there is another way to change the relative phase of our computer bits.

While we can't do this under computer control, we can manually adjust both the Apple II video circuit and the color television set so as to change the phase of the color reference signal itself. The Apple II control is labeled *color trim;* the television set's control for this is usually called *tint* or



The Only Computer Reference Book You Need.

The Only Reference Book You'll Ever Need.

The only book that gives you all the ingredients for a successful computer system. Interested in the Apple II?™ You'll need an FCC-approved RF modulator to connect it to your color television set. We tell what they are, where to get one, and how to hook it up.

Thinking of using Radio Shack's TRS-80™ for a low-cost word processing system? You'll need an upper-case/lower case modification for the keyboard. We'll tell you what it involves, where to get it done, or how to do it yourself.

Software? We've sampled canned software and will tell you what suits our taste. Rather do it yourself? We have algorithms in our machine-independent "Cookbook" language for everything from fast sorts to phototypesetter hyphenation. Plus a crash course in BASIC for those who know FORTRAN. And much, much more.

Last but not least: The Cookbook Yellow Pages, a complete guide to small computer services and products. We update The Yellow Pages every two months—and send them to you free with the latest additions, corrections, and improvements of our white pages. You just plug them into our loose-leaf binder.

Get with The Cookbook. The Best is Yet to Come.

Send me The Cookbook. I enclose $15 plus $1.50 for shipping, which will be by UPS unless I've marked this form U.S. mail (____check here). California residents add $.98 sales tax. Allow up to four weeks for possible out-of-stock conditions. Make checks payable to "The Computer Cookbook". Mastercharge and VISA accepted with 13-digit card number and 4-digit expiration date.

Name ____________________

Street ____________________

City ____________________ State ____________ Zip ____________

Mail to: The Computer Cookbook / P.O. Box 4084 / Berkeley, CA 94704

Circle 52 on inquiry card.

hue. The combined range of adjustment of these two controls is usually enough to go at least halfway around the color circle of figure 2, putting one or the other of our complementary pairs of colors at any point on the circle. Thus we can adjust for any pair of complementary colors we want: blue and yellow, green-blue and orange, cyan and red, green and magenta, or yellow-green and violet. So long as we don't require the ultimate in horizontal resolution, we can have any two complementary colors plus black and white for our high resolution graphics using only ones and zeroes as data. If the colors listed above and in figure 2 don't seem exactly complementary, it's largely because of the broad range of hues to which we carelessly apply the name *blue.* If we let the television picture-tube phosphors define our red, green, and blue, then the complementary colors are those of figure 2. The television set is adjusted such that red + green + blue = white. Since complementary pairs also add together to give white, it follows that the sum of any two of the three primaries gives the complement of the third: for example, the complement of red is actually green + blue, or cyan.

Extra Colors

Studying the Apple II specifications in the light of the National Television Systems Committee color standards led me to expect it to work this way, but that isn't quite the end of the story. After I got my Apple II, as I was casually watching the random sine wave program on the high resolution demonstration tape, there in living color was a display with *four* colors. After a bit of head scratching and experimenting with the adjustments on my portable color set, I think I have the explanation.

First of all, the single dot patterns give the two complementary colors, just like it says in the script. Alternating double dots, which ought to be displayed as black and white, actually show up as a weaker version of the same pair of complementary colors if the television set is adjusted normally, that is, with the fine tuning just backed off from the setting that first produces sound bars in the picture. But if I back the fine tuning farther away from this setting (any automatic fine tuning or tint controls should be switched off), just before the color signal drops out, the weak colors on the double dot patterns brighten and shift to another pair of complementary colors. The exact colors depend upon the setting of the tint control, but they are more than 30 degrees from the first pair, so if the single dot patterns give red and green, for example, the double dot patterns appear as orange and blue.

It's hard to figure out how the double dot patterns get displayed in color since they are square waves at half the color frequency and ought to contain a zero component at 3.58 MHz. Apparently the video detector circuit in the set produces enough second harmonic distortion to activate the color circuits. Mistuning puts this signal near the cutoff of the color bandpass filter where there is maximum phase distortion. I tried this out on the more expensive television set at the store where I bought my Apple II, and although it's more difficult to get the adjustments just right, the extra colors are there. Ironically enough, this trick seems to work better on cheap sets.

So there you have it. Whether you prefer colors or resolution, the Apple II high resolution graphics will put out all you can get through the antenna terminals of a color television set with just different patterns of ones and zeroes. To find out what your set will do, you need to display vertical lines with the single dot and double dot patterns. An easy way to do this is to load the Apple II high resolution demonstration tape and select the program that sums two sine waves. When the program asks for two frequencies, enter 63 and 64 to get the pattern shown in photo 1. Other numbers you may want to try are combinations of 31, 32, 33, 63, 64,

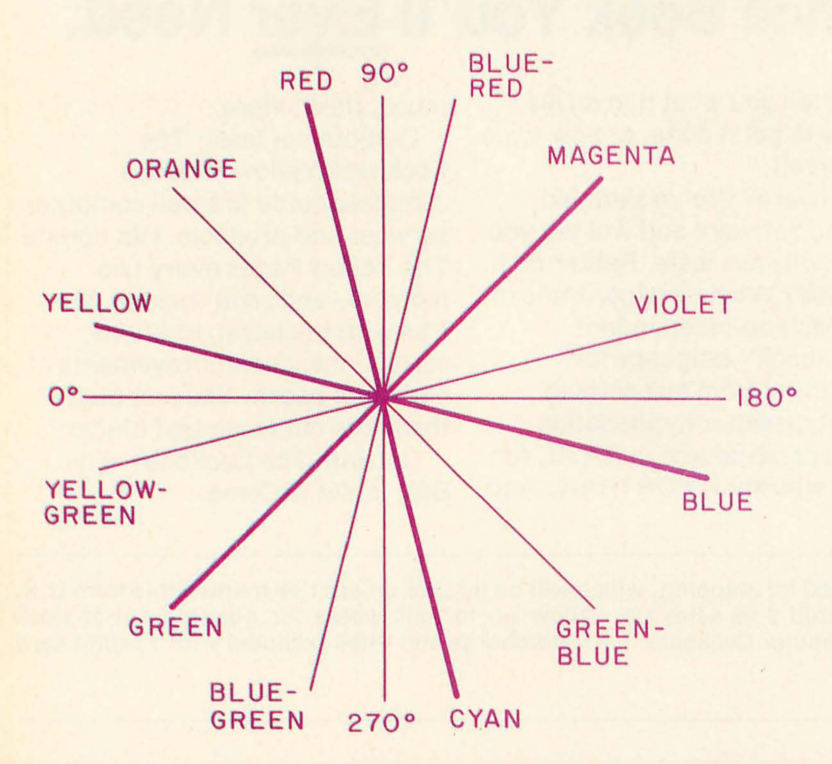


Figure 2: Arrangement of possible colors versus signal phase. The Apple II high resolution graphics outputs two complementary colors (colors that are separated by 180 degrees on the circle).

The Intecolor 8070 business system. At twice the price, it would still be inexpensive.



Unretouched photo of screen

Furniture not included

*U.S. domestic prices

ISC SALES REPRESENTATIVES: AL: 205/883-8660, **AK:** (GA) 404/449-5961, **AZ:** 602/994-5400, **AR:** (TX) 214/840-2169, **CA:** Alhambra 213/281-2280, Goleta 805/964-8751, Irvine 714/557-4460, Los Angeles 213/476-1241, Mountain View 415/964-9300, San Diego 714/292-8525, **CO:** 303/759-0809, **CT:** (GA) 404/449-5961, **DE:** (GA) 404/449-5961, **DC:** (VA) 703/569-1502, **FL:** Orlando 305/425-5505, Ft. Lauderdale 305/776-4800, Melbourne 305/723-0766, Tallahassee 904/878-6642, **GA:** 404/455-1035, **HI:** 808/524-8633, **ID:** (UT) 801/973-7969, **IL:** (No.) 312/564-5440, (So.) (MO) 816/765-3337, **IN:** (IL) 312/564-5440, **IA:** (MO) 816/765-3337, **KS:** (MO) 816/765-3337, **KY:** 606/273-3771, **LA:** 504/626-9701, **ME:** (GA) 404/449-5961, **MD:** (VA) 703/569-1502, **MA:** (GA) 404/449-5961, **MI:** 313/227-7067, **MN:** 612/822-2119, **MS:** (AL) 205/883-8660, **MO:** 816/765-3337, **MT:** (CO) 303/759-0809, **NB:** (MO) 816/765-3337, **NH:** (GA) 404/449-5961, **NJ:** (GA) 404/449-5961, **NV:** (AZ) 602/994-5400, **NM:** 505/292-1212, **NY:** (GA) 404/449-5961, **NC:** 919/682-2383, **ND:** (MN) 612/822-2119, **OH:** Dayton 513/429-9040, Cleveland 216/464-8113, Columbus 614/436-2051, **OK:** (TX) 214/840-2169, **OR:** 503/620-5800, **PA:** 412/922-5110, **RI:** (GA) 404/449-5961, **SC:** 803/798-8070, **SD:** (MN) 612/822-2119, **TN:** 615/482-5761, **TX:** 214/840-2169, El Paso Area (Las Cruces, NM) 505/523-0601, Houston Only 713/780-2511, **UT:** 801/973-7969, **VT:** (GA) 404/449-5961, **VA:** 703/569-1502, **WA:** 206/455-9180, **WV:** 412/922-5110, **WI:** (IL) 312/564-5440, **WY:** (CO) 303/759-0809, **EUROPEAN EXPORT SALES**: **EUROPE**: (MA) 617/661-9424, **BELGIUM**: Brussels 02-242 36-04, **FRANCE**: Rueil Malmaison 749-40-37, **GREECE**: Athens 642-1368, **ITALY**: Roma 805-647/872-457, **THE NETHERLANDS**: Poeldijk 01749-7640, **SPAIN**: Barcelona 204 17 43, **SWEDEN**: Vallingby 08-380-370, **SWITZERLAND**: Mutschellen 057-54655, **UNITED KINGDOM**: Bournemouth 0202-293-115, **WEST GERMANY**: München 089-31881, **AUSTRALIA & NEW ZEALAND**: Melbourne 543-2077, Meadowbank 808-1444, Chermside 59-6436, Wellington 64-4585, Auckland 876-570, **CANADA**: Datamex, Ltd. (Distributor) Dorval 514/636-9774, Ottawa 613/224-1391, Toronto 416/787-1208, Vancouver 604/684-8625, **CENTRAL & SOUTH AMERICA & CARIBBEAN**: (GA) 404/394-9603, **MEXICO**: Monterrey 564-876, **FAR EAST**: (CA) 213/382-1107, **HONG KONG**: 5-742211, **JAPAN**: Tokyo 402-8596, **TAIWAN**: Taipei 02-7026284, **MIDDLE EAST**: **IRAN**: Tehran 891148, **ISRAEL**: Tel Aviv 266-291, **KUWAIT**: Kuwait 438 180/1/2, **LEBANON**: Beirut 221731 260110, **SAUDI ARABIA**: Jeddah 27790, Ryadh 25083-39732, **UNITED ARAB EMIRATES**: Sharjah 24068.
For sales and service in other countries contact ISC headquarters in Norcross, GA., U.S.A.

ISC **Intelligent Systems Corp.** □ 5965 Peachtree Corners East □ Norcross, GA 30071 □ Telephone 404-449-5961 □ TWX 810-766-1581

65, and 95. Apparently there is a lot of sampling error when the frequencies you select don't fit the table the program uses to generate the sine waves. If you experiment until you find the limits of your particular television set, you'll know how to make high resolution pictures on your Apple II in just about any colors you want.

BIBLIOGRAPHY

1. *Apple II Reference Manual,* Apple Computer Inc, Cupertino CA, 1978.
2. Herrick, Clyde N, *Color Television: Theory and Servicing,* Reston Publishing Co, Reston VA, 1973.
3. Schure, Alexander, *Basic Television,* revised second edition, volume 6, Hayden Book Co, Inc, Rochelle Park NJ, 1975.

Addendum

The following comments were received from Steve Wozniak of Apple Computers:

Thank you for passing along Allen Watson's article on the Apple II high resolution colors.

As Allen discussed, Apple II high resolution colors are the result of alternating zeroes and ones on the screen. The exact colors generated depend on the phase (or timing) relationship between the display signal and the color reference phase. By adjusting the television controls, any desired color pair may be displayed.

Oddly enough, only the seven least significant bits of the Apple II high resolution refresh memory bytes are used (examples are shown in figure 1). A simple modification allows the high order bit of each to specify one of two color sets by generating a 90 degree phase shift of displayed information. (Yet more colors may now be obtained by applying the technique suggested by Allen.)

Adding the High Order Bit Modification to the Apple II

1. Remove the Apple II printed circuit board from its enclosure.

(a) Remove the ten screws securing the plastic top piece to the metal bottom plate. Six of these are flat head screws around the perimeter of the bottom plate and four are round head screws located at the front lip of the computer. All are removed with a Phillips head screwdriver. Do not remove the screws securing the power supply or nylon insulating standoffs.

(b) Lift the plastic top piece from the bottom plate while taking care not to damage the ribbon cable connecting the keyboard to the printed circuit board. This cable will have to be disconnected from one or the other.

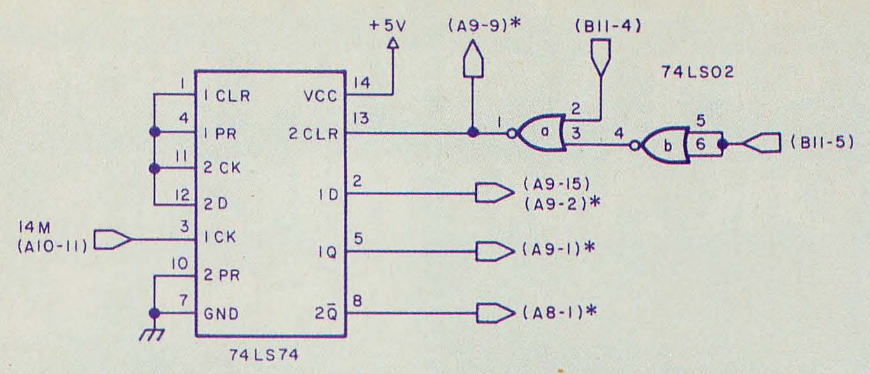


*Figure 3: Circuit which must be added to the Apple II to add more colors to the high resolution display. (*Caution*: Adding this circuit voids the warranty.) A starred assignment (*) indicates that the connection is made to a pin which is out of its normal socket. Besides the connections which are shown, also connect pin (B8-14) to (A8-6) which is out of its socket, and (B8-7) to (A8-13) which has also been removed from its socket. The power connections to the 74LS02 are +5 V to pin 14 and ground connected to pin 7.*

(c) Disconnect the power supply from the printed circuit board.

(d) Remove the #8 nut and lockwasher securing the center of the printed circuit board. These will not be found on the earlier Apple II computers.

(e) Carefully disengage each of six nylon insulating standoffs from the printed circuit board (seven on earlier versions).

(f) Lift the printed circuit board from the bottom plate.

2. Above the board wiring method.

(a) Lift the following IC (integrated circuit) pins from their sockets.

IC	Pin Number
A8	1
A8	6
A8	13
A9	1
A9	2
A9	9

(b) Mount a 74LS74 (dual C-D flip-flop) and a 74LS02 (quad NOR gate) in the Apple II breadboard area (A11 to A14 region).

(c) Wire the circuit in figure 3.

3. Below the board wiring method.

(a) Desolder all pins of socket A8. Lift the socket and its 74LS257 integrated circuit off the printed circuit board taking care not to destroy it. Cut the trace between pins 6 and 13 of A8 on the top side of the board. Also cut the trace between pins 13 and 15 on the top. Reinsert socket A8 and the 74LS257. *Be careful.*

(b) Cut traces going to the following pins on the bottom of the Apple II board. Each pin should have a single trace going to it. *Be careful.*

IC	Pin Number	IC	Pin Number
A8	1	A9	1
A8	6	A9	2
A8	13	A9	9

(c) Connect pin 15 of ICA8 to ground (pin 8 of ICA7 on the keyboard socket is a nearby ground).

(d) Mount the 74LS74 and 74LS02 as per step (b) of the above the board wiring method.

(e) Wire the circuit of the above the board wiring method, step (c). All wires are on the bottom of the Apple II board and no pins need be removed from their sockets or soldered to.

4. Reassemble the Apple II and make sure it is operational. If not, check all wiring very carefully. Make sure that all integrated circuits are in their sockets and properly oriented.

5. The following color values are now applicable to the high resolution subroutines:

BLACK2	128
ORANGE	170
BLUE	213
WHITE2	255

For example, the program below draws an orange line from location (10, 20) to (200, 140). It is assumed that the high resolution routines are already in memory locations hexadecimal 800 thru BFF.

```
 0 X0 = Y0 = COLR
 5 INIT = 2048 : PLOT = 2830 : LINE = 2836
 7 ORANGE = 170 : CALL INIT
10 X0 = 10 : Y0 = 20 : COLR = ORANGE :
   CALL PLOT
20 X0 = 200 : Y0 = 140 : CALL LINE
30 END ■
```

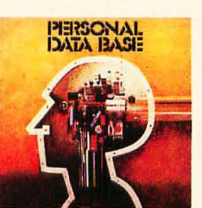
Look how soft you can have it with a Compucolor II® personal computer.

STAR TREK (*ST)
Star Trek, Lunar Lander, Shoot, and Tic-Tac-Toe.

SAMPLER (*SA)
Biorhythms, Concentration, One-Armed Bandit, 3 more.



HANGMAN (*HA)
Hangman, Math Tutor, Two-To-Ten.

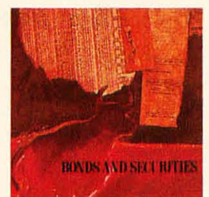


PERSONAL DATA BASE (PD) Personal Data Base. File and data management system. $29.95

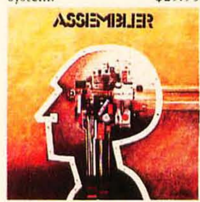


FORMATTED SOF-DISK (*FD)
2-pack Sof-Disks for your own BASIC programs.

MATH TUTOR (*MA)
Math Tutor, Checkbook, Recipes, 2 more.

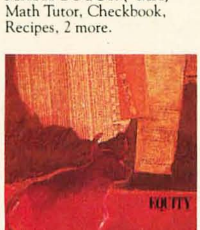
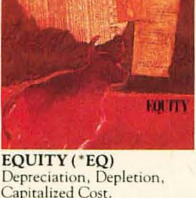


BONDS & SECURITIES (*BO)
Bond Price & Yield, Amortization Schedule, 2 more.



ASSEMBLER (AS)
Assembler. Assembles 8080 Assembly Language programs. $24.95

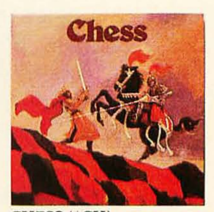
TEXT EDITOR (TE)
Text Editor. Facilitates program writing and correcting. $24.95

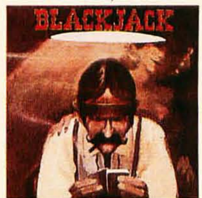
EQUITY (*EQ)
Depreciation, Depletion, Capitalized Cost.



CUBIC TIC-TAC-TOE (*CU)
Cubic Tic-Tac-Toe, Swarms, 2 more 16K programs.

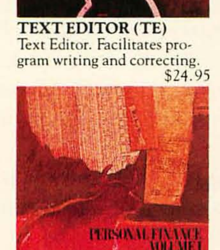


CHESS (*CH)
Chess, Acey-Deucey, Line Five, Biorhythms.

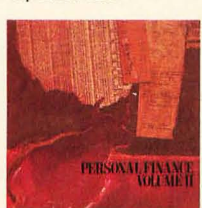


BLACKJACK (*BL)
Blackjack, Roulette, Slot Machine, 2 more.

OTHELLO (*OT)
Othello®, Math Dice, 2 more. "Othello" is the registered trademark of CBS Inc. for its strategy disc game.



PERSONAL FINANCE, VOL. I (*PI)
Rate of Return, Interest Conversions, 2 more.



PERSONAL FINANCE, VOL. II (*PII)
Loans: Sinking Fund, Declining Interest, Constant Ratio.

When you have a Compucolor II — the exciting color personal computer with a built-in mini-disk drive — you really have it soft. $19.95 is all you pay for most Compucolor Sof-Disk™ albums.* They're formatted to run only on the Compucolor II, and each contains at least three programs. That's far better value and variety than you get with programmed cassette tapes. And the Sof-Disks you see here are just a sampling. New introductions are added each month.

To start your collection of mind-blowing games and other useful programs, visit your nearest computer store. Or, if there's no dealer in your area, clip the coupon and mail your order today.

First things first! If you'd like more information, simply check the coupon for a free brochure about the Compucolor II.



YOUR CHOICE $19.95*

To order your Sof-Disk albums, simply enter the album(s) of your choice, by letter(s), in the spaces provided. If you want to order more than one copy, please enter the number of copies desired beside the letter. Add $.50 for each album ordered to cover costs of mailing and shipping. Georgia residents add sales tax. Mail with check or money order to address below.

*See ad for varying prices on Assembler, Text Editor, and Personal Data Base Sof-Disks.

Name ______________________

Street ______________________

City ______________ State ________ Zip ________

☐ Send me the name of my nearest Compucolor Dealer.
☐ Please tell me more about the Compucolor II!
☐ I'd like additional software information!

Mail to: Compucolor Corporation
Dept. B
P.O. Box 569
Norcross, Georgia 30071

Allow 3 weeks for delivery.

Circle 51 on inquiry card.

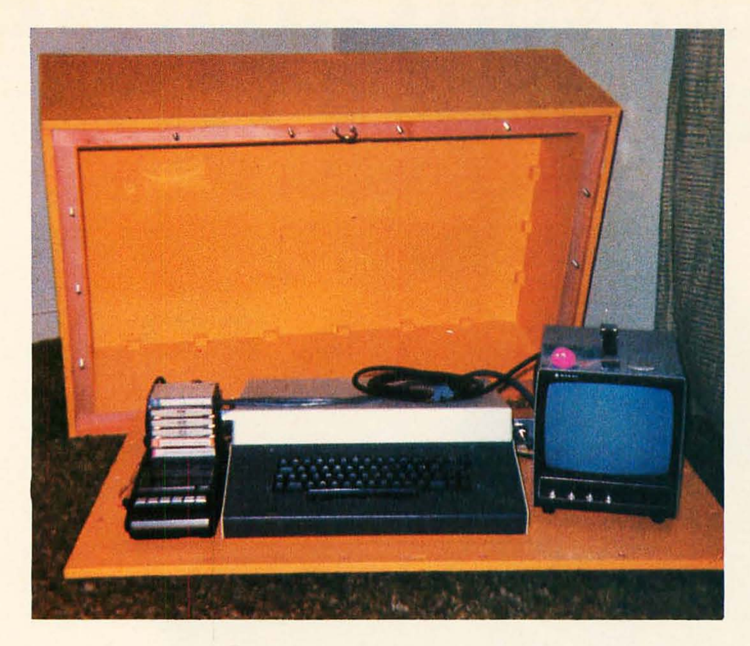


Photo 1: The author's homemade computer system cabinet with top removed.

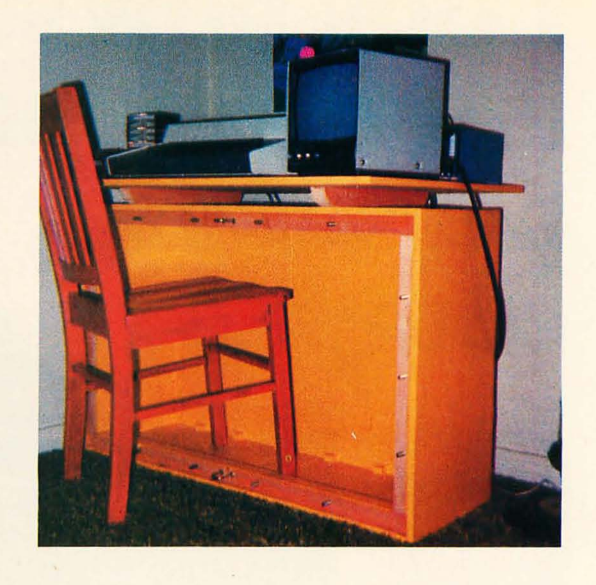


Photo 2: The system as it appears when converted into a desk. Note that the floor of the cabinet becomes a desk top displaying the entire system ready to plug in and run.

A Home for Your Computer

Joseph Dawes
2510 Broadway
Big Spring TX 79720

If you're good at woodworking, here's a project that will save you a lot of time and trouble: build a cabinet for your computer! A carefully planned cabinet, as any ham or hi fi buff knows, serves to increase the utility and enjoyability of the equipment inside it.

I started planning my computer system cabinet as soon as I scattered my components around the cabinet I had made for my amateur radio gear. First of all, the cassette unit had to be fastened down: I quickly tired of holding it in place with one hand while unplugging something or changing a cassette. The power supply was constantly running warm and I knew it should be on stilts to increase heat loss. My separate video monitor could have ended up either beside the processor-keyboard cabinet or on top of it, but I quickly decided to reserve the top of the processor cabinet for a desk. The monitor would be beside the processor but angled toward the operating position for improved visibility. The keyboard had to be at a comfortable typing height, somewhere from 28 to 30 inches from the floor.

With these parameters in mind, I had to decide on cabinet style. The styles that first came to mind were the living room furniture piece with finely grained wood and the hobby room piece with modest wood grain or painted wood. However, circumstances led me to develop a somewhat different cabinet style. I hope my final design will prompt readers to forge ahead with their own cabinet ideas.

The need to transport the system outside my home, combined with my dislike for connecting and disconnecting wires, dictated the cabinet design shown in figure 1 and photos 1 through 3. If I wanted to demonstrate it at the school where I teach or elsewhere, it would have to be operable as soon as it was opened and plugged in. Nothing kills interest more than 30 minutes of wire fiddling.

The cabinet shown is sturdy enough to take some licks in a truck or car. While it is closed and bolted, a bicycle chain can be run through the two U-bolts and around the nearest oak tree, making it very inconvenient to move or to open without some commotion. When the lid is taken off the equipment base, it can be turned up on one side to become a pedestal upon which the equipment base can be set. The whole affair is quite stable when set upon a reasonably level surface, and the lid interior provides knee room and space for keeping notebooks and demolished programs. When closed it can be sat upon, and, although there's absolutely no way to lug it around in a VW beetle, it will fit in the back (not the trunk) of my 2 door Falcon with a little imaginative stevedoring. In short it does what I require very nicely, and if I had to

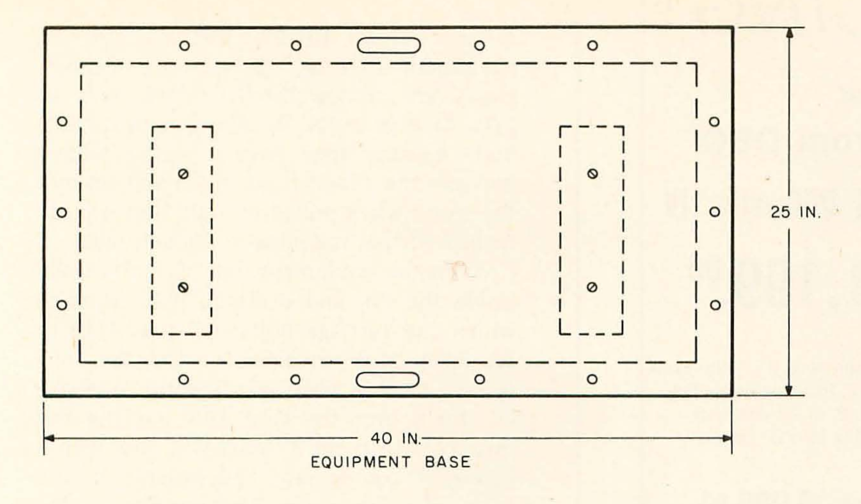
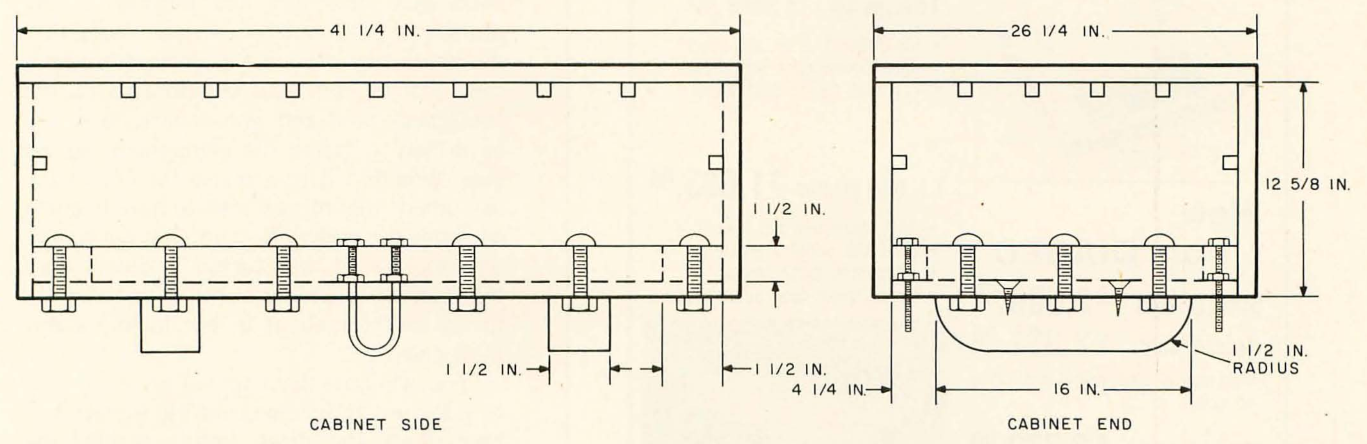


Figure 1: Plans for building the author's computer system cabinet. The unit is built from 5/8 inch plywood and features two U-bolts for adding an antitheft chain. All components are mounted on the floor of the cabinet. See accompanying photos.



build it over again I can think of very little I would change. After I painted the inside of the lid, it looked fine in a corner of the living room. What more could one ask?

Some comments on construction details might be helpful. The size of the equipment base is dictated strictly by the dimensions of the equipment to be placed on it. The width of the base, however, should not be much wider than 25 inches or the work surface will be too high for operating comfort when the cabinet is open (unless you like to stand while you work). Note that all equipment must fit within the dotted line perimeter shown in the equipment base drawing. Otherwise, equipment may get damaged by the bolt rails! I cut the plywood pieces for the lid with a smooth cutting plywood blade in my table saw and took great care to make cuts exactly to dimension, because the whole lid was assembled by gluing the side pieces to the top. The small gluing blocks are tack-nailed in place while the glue dries. These blocks are for structural reinforcement and *do not* hold the lid together while the glue sets. Bar or pipe clamps must be used to provide the necessary pressure. A cabinet-making friend or a school shop might be a good source for these if you have none. The only permanent metal fasteners used are the two screws through the base into each runner, and they serve primarily as gluing clamps as well.

The bolt rails are glued around the inside edge of the lid (Be certain to recess them a

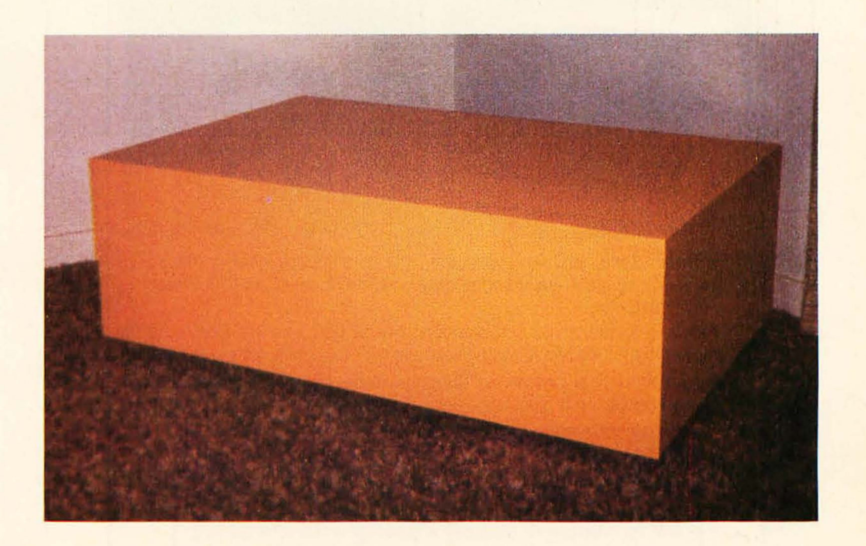


Photo 3: The cabinet locked up and ready for transport.

distance equal to the thickness of the equipment base as the drawing shows!). Holes are drilled through these rails to take carriage bolts. Be sure to use carriage bolts because they have a square section beneath the round head which will set into the wood when pulled up tight and will not require further fastening to the bolt rails.

Trim the equipment base so it fits easily inside the lid, and drill the holes through which the carriage bolts will pass 1/16 or 1/8 inch larger in diameter than the bolts to save trouble when you first try to set the lid down over the base. Cut out the slot through which the U-bolts will pass with a jigsaw or coping saw, and provide a little extra play here too. The position of the runners should not vary much from that shown, but before you decide exactly where they will be, set your components on the equipment base and note where holes will be drilled to fasten the components to the base. Wouldn't it be a shame for one of the equipment mounting screws to have to come up from the underside right where a runner was located? I was lucky! I didn't even think about this hazard until it was too late to do anything about it, but luckily everything cleared.

You will note that edges of plywood tend to have unsightly gaps, and the surface will have dents and dings. Before sanding the wood preparatory to painting, take a putty knife and some patching paste and fill these defects thoroughly. It's easy to do and makes all the difference in the finished job. When you sand, these fillings will level off, corners and edges will get smoothed and slightly rounded, and you'll get a smoothly painted surface. I avoid painting whenever I can, but when painting raw plywood I have to admit that the final results are well worth the trouble of first putting on a primer coat and then a finish coat with a light sanding between coats. The only thing worse than painting once is painting twice, but when it's all over you have a surface you don't mind showing off.

As a finishing touch, some acorn nuts look good on the ends of the carriage bolts since they protrude toward the operator when the cabinet is set up.

For icing on the cake how would you like to hear that this whole project can be cut out of one sheet of plywood? Tough luck! Unless someone out there pulls a topological trick or is the owner of a more compact system than mine, you'll need a whole sheet and a scrap from the lumber company's cutoff pile for one of the ends. Sorry about that, but everything doesn't always work out for the best.

Happy woodworking! ■

**Tinker, Tailor, Soldier, Sailor . . .
Doctor, Lawyer . . . the Chieftain's here.**

No matter whether you're a serious hobbyist or a serious businessman, the Chieftain 6800 microcomputer with capabilities that surpass the Z-80 is made for you.

Smoke Signal's quality-packed Chieftain I features two 5.25-inch mini-floppy drives and Chieftain II features two 8-inch floppy drives.

Both microcomputers provide 32K static memory, two serial I/O ports, a 2 MHz processor board, a 2K RAM monitor, a nine-slot motherboard with built-in baud rate generator and gold connectors for high reliability. The Chieftain's stylish leather-grained cabinet houses the above with its own cooling fan and regulated power supply.

Every Chieftain is complete with system software and is totally burned-in as well as tested to further insure high reliability.

And it's expandable to 64K memory with up to 2 megabytes floppy disk storage.

So see your nearest Smoke Signal dealer, he'll be glad to show you how to get your wampum's worth. Systems start at $2,595.



Hail to the Chieftain

Smoke Signal Broadcasting, 31336 Via Colinas, Westlake Village, CA 91361, (213) 889-9340

Circle 328 on inquiry card.

Talk to a Turtle

Build a Computer Controlled Robot

James A Gupton Jr
7416-G Pebblestone Dr
Charlotte NC 28212

What do personal computer experimenters do with their microcomputers when computer games lose their appeal and they tire of programming things like, "140 FOR X = 1 to 500: PRINT X: NEXT X "? The exciting idea of adding a computer controlled robot suggests building your own R2-D2 robot from *Star Wars.* It might not be wise to start with a project as sophisticated as duplicating R2-D2, but there *is* a way you can begin a robot project on a smaller scale. It works, too!

The Terrapin Turtle is a fascinating robot project that most experimenters can fully assemble in four hours. It runs forward, backward, turns left or right, blinks light emitting diode eyes, and can talk in a two tone beep. Its shell is mounted on a spindle that engages one of four microswitches. These relay a signal back to the computer

ABOUT THE AUTHOR:

Mr James Gupton Jr has a most unusual background including photography, electro-optics research and development (which resulted in five patents on computer video display tubes and phosphor screens), along with teaching electronics. The Union County Career Center is the only high school in North Carolina to provide an electronics program which covers subjects from direct current to microprocessors. This program is under the guidance of Mr Gupton.

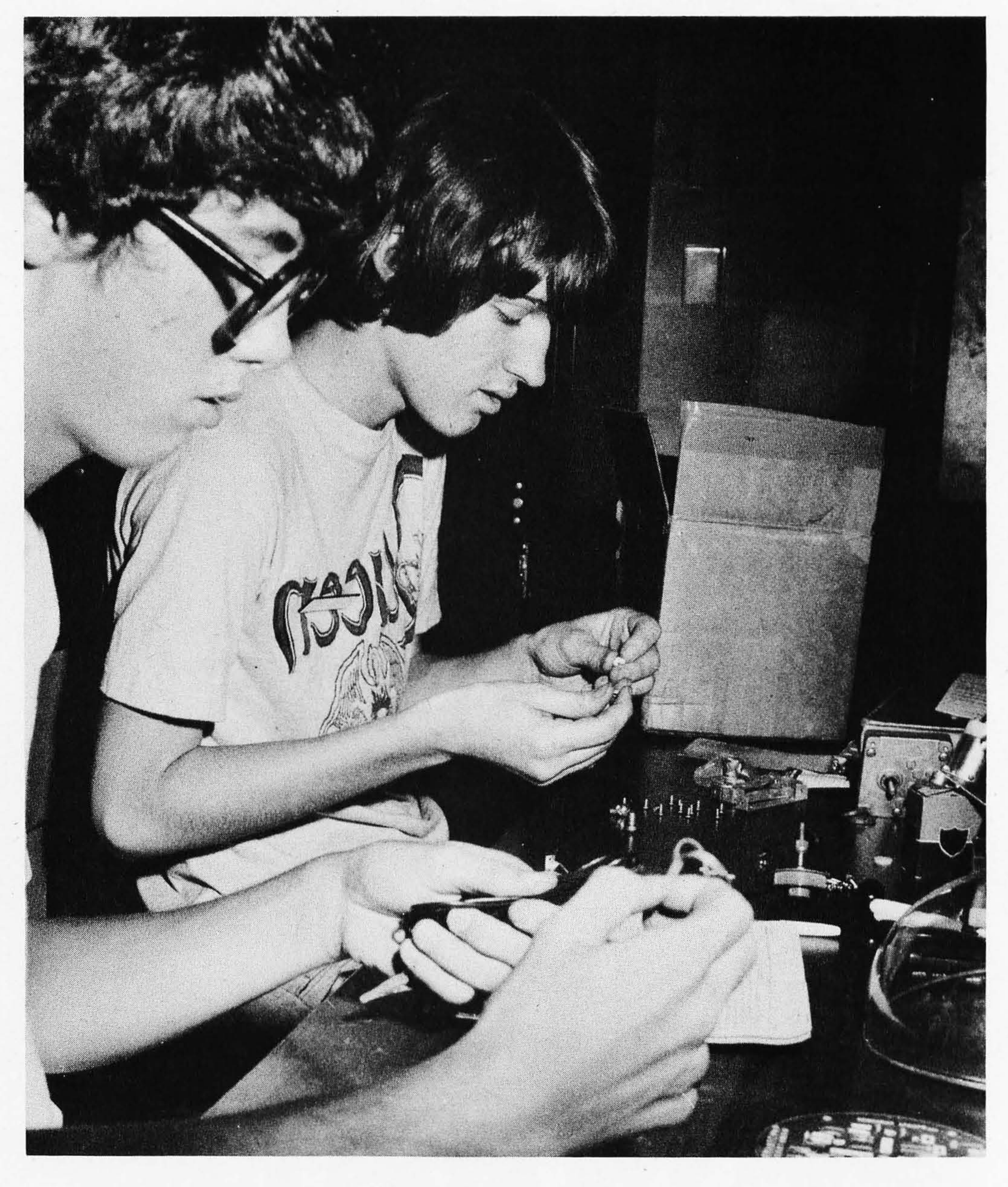


Photo 1: Jeffrey Dunn (foreground) and Richard Voss check off the Turtle components against the parts list.

over its 10 foot umbilical cord, indicating when the Turtle has run into something from either front, right, left, or rear side. If you direct the Turtle on an exploratory trip around the room, its journey can be recorded by your microcomputer. On completing its journey, the Turtle can actually draw a map of its path using an internal ball point pen.

The Terrapin Turtle illustrated in this article was assembled by high school students at the Union County Career Center in North Carolina. The total assembly time was four hours from start to initial test. This article is not intended for use as a construction project, but rather to introduce you to computer controlled robots.

Assembling the Terrapin Turtle

The cardinal rule for assembling any electronic kit is to begin by checking off each component on the parts list. Photo 1 shows Jeffrey Dunn and Richard Voss checking the components of the Turtle kit

Resistors

510 ohm ¼ W: R9, R10
100 ohm ¼ W: R21, R30
15 K ohm ¼ W: R5, R6, R7, R8, R29, R22
50 K ohm ¼ W: R19, R20
1 K ohm potentiometers: P1, P2, P3, P4
1 K ohm ¼ W: R1, R2, R3, R4, R11, R12, R13, R14, R15, R16, R17, R18, R23, R24, R25, R26, R27, R28

Capacitors

C1	0.1 mF 35 V
C2, C3	500 mF 35 V

Diodes

1N4000	D1, D2, D3, D4, D5, D6, D7, D8, D9
3.9 V zener	D10

Transistors

2N2222	Q1, Q2, Q3, Q4, Q5, Q6, Q7
GE-D40C4	Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q19

Table 1: The Turtle component part list. The complete Turtle kit, including all hardware, printed circuits, electronic components is available from Terrapin Inc for $300.

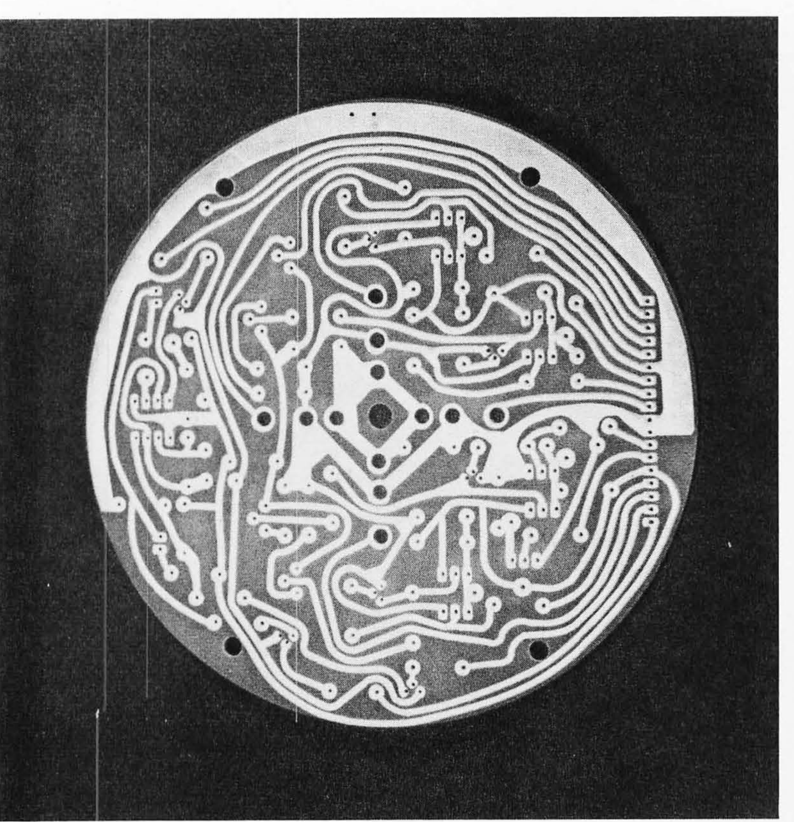


Photo 2: The foil side of the Turtle's printed circuit board.

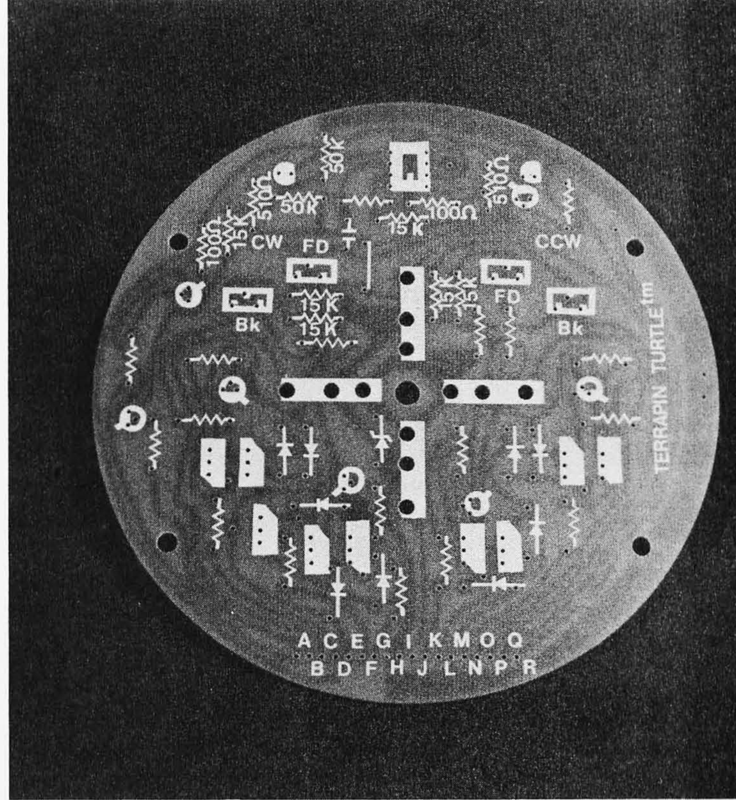


Photo 3: The component side of the circuit board.

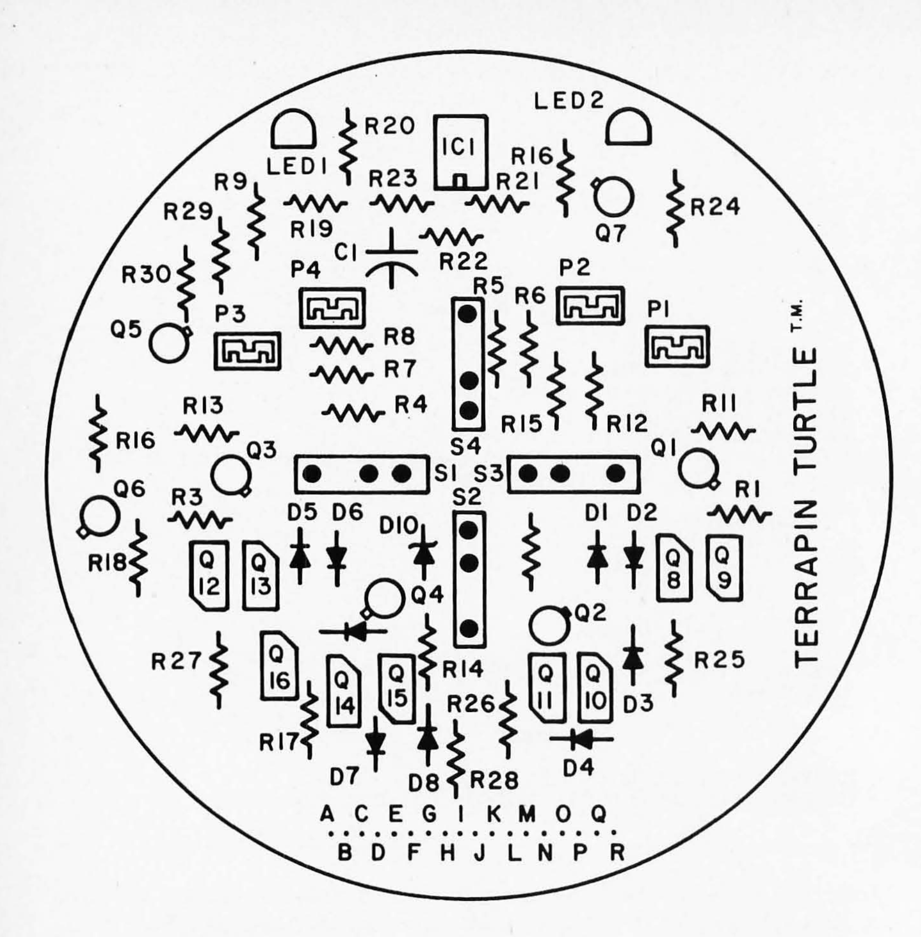
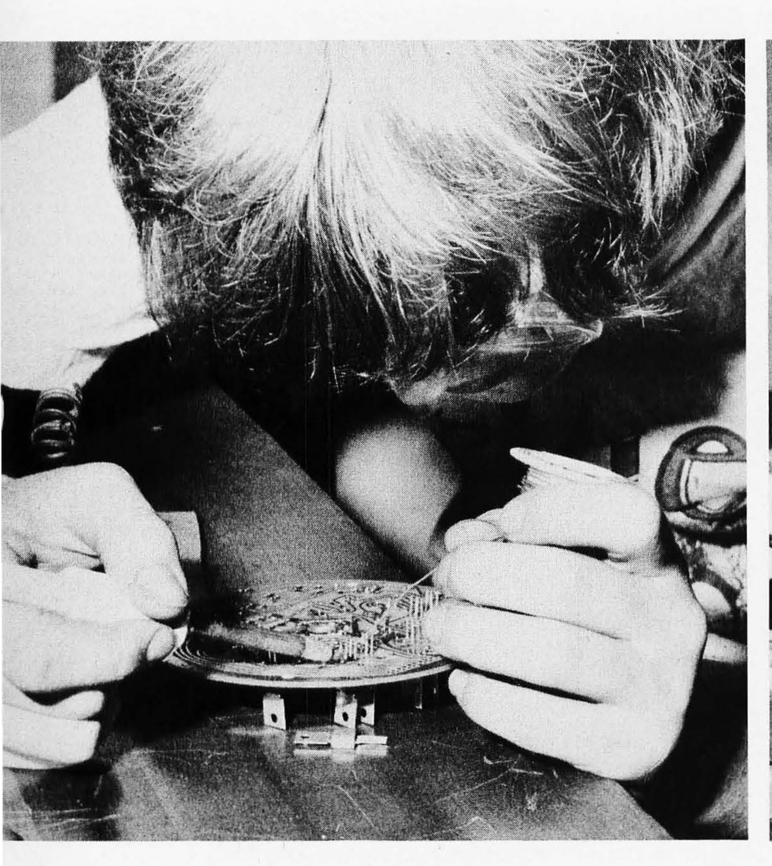
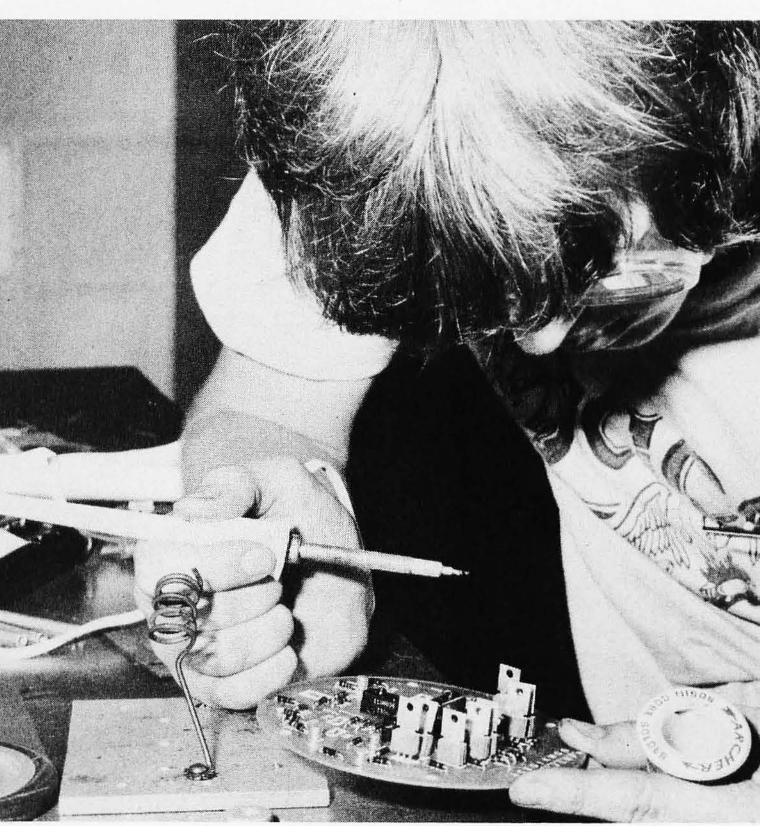


Figure 1: Supplemental diagram showing component identification numbers which relate to schematic locations.

against the parts list. Once assured that everything was included in the kit, the printed circuit board was examined for possible scratches. Photo 2 shows the etch side of the 5 inch diameter printed circuit board. Photo 3 shows the component side. It was quickly noted that not all resistor values were printed on the component side, and that there was no identifying resistor number to relate any resistor to the schematic. The instruction booklet stated that eighteen 1000 ohm resistors should be placed where the resistor symbols did not have a value indicated. Figure 1 is a supplemental instruction that identifies each component corresponding to the schematic diagram.

Richard Voss was in charge of assembling the printed circuit board for the Union County Career Center's Turtle. Photo 4 shows the soldering of the Darlington transistors that control both of the Turtle's drive motors. Notice the micro-tip, low wattage soldering iron and 0.020 inch (0.05 cm) diameter solder being used. All too frequently electronic kits are damaged during assembly by the use of high wattage soldering tools which damage the heat sensitive foil and apply too much solder. An excess of solder can short out both the closely spaced component pads and the circuit paths with solder bridges. Once the soldering has been


Photo 4: Soldering the installed components.


Photo 5: Inspecting the assembled components.

finished, it is wise to recheck the placement of the components, just to be doubly sure.

Care must be exercised to keep the tabs on the Darlington transistors from touching one another. A small piece of plastic tape on each tab will save the transistor should the tabs accidentally be brought together while under power. Photo 5 shows the final inspection of the assembly of components onto the printed circuit board. Photo 6 shows the completed circuit board.

Figure 2 provides the circuit schematic for the control of the Turtle's left and right motors and the internal ball point pen. The pen is lowered by a 12 V solenoid upon command from the computer. Figure 3 shows the schematic for shell touch sensors, lights, and sound control. The figure also shows the power attachment points for the operation of the Turtle's electronics and motors. A 12 V, 3 A power source is required for the best performance. The Turtle can operate, however, with a power source of 1 A capacity if the 3 A source is not available. The Turtle illustrated in this article was powered by a 4 A regulated power supply.

Photo 7 shows the final assembly of the printed circuit board onto the motor housing. The most difficult part of the entire assembly was forcing the rubber tires onto the wheels. It is almost impossible to do this

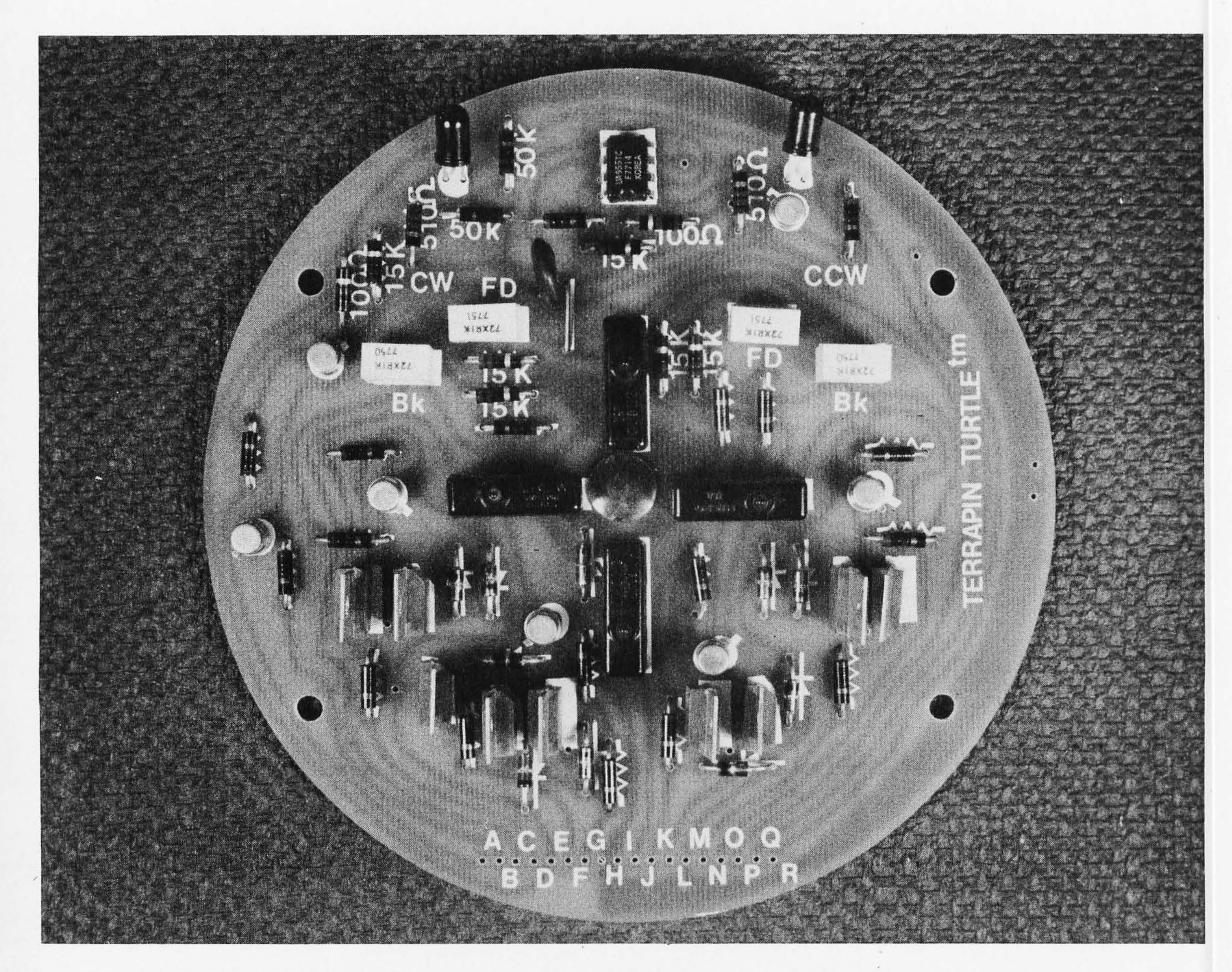


Photo 6: The completed board, showing the uncluttered layout.



Photo 7: The printed circuit board, shown attached to the Turtle's motor drive housing.

by yourself — a second set of hands will be needed to mount the rubber tire onto the wheel. Photo 8 shows the assembled Turtle minus its sensor shell and the two power supplies used for testing without the use of a microcomputer. The Turtle is controlled with a TTL (transistor-transistor logic) voltage of 0 V and +2 V. This may cause some problems for parallel interfaces that function between 0 V and +5 V. The higher voltages can damage the 2N2222 Darlington tran-

Text continued on page 84

Photo 8: The completed Turtle, connected to a power supply for testing.

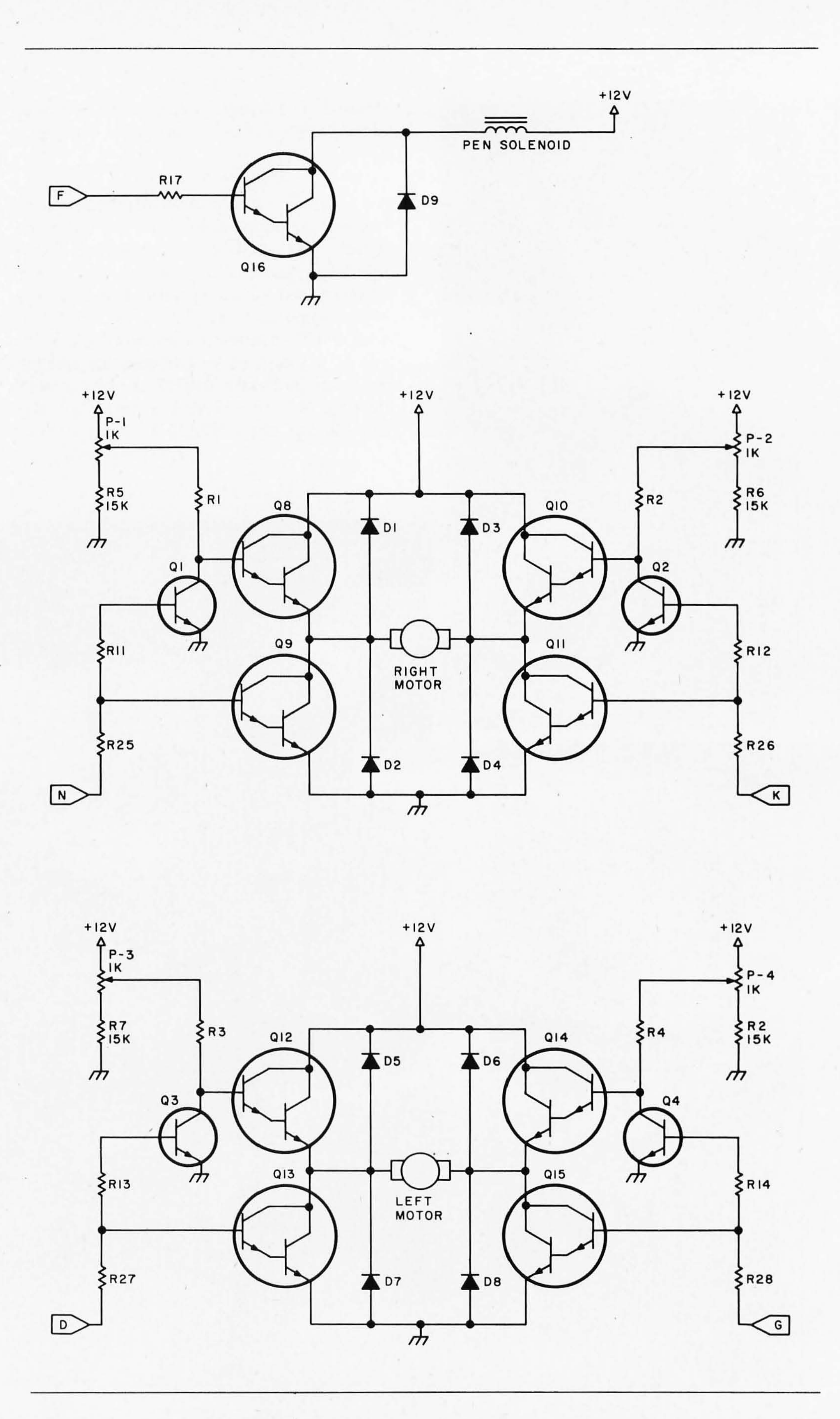
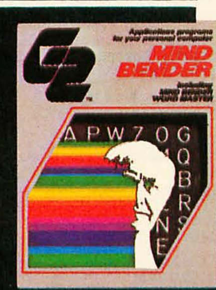


Figure 2: Schematic of the Turtle's motor control and pen control circuitry.

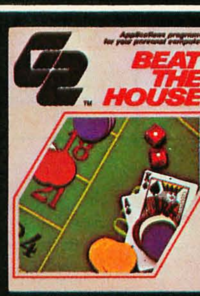
FUN & GAMES



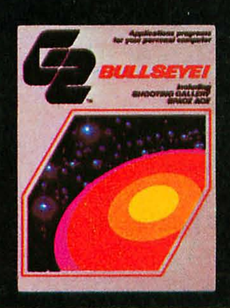




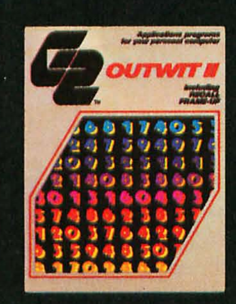


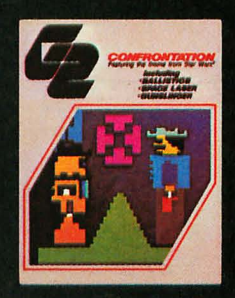


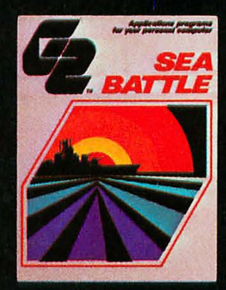






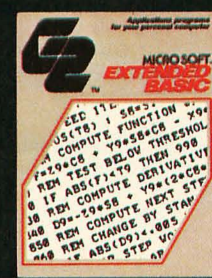






PLUS SERIOUS SYSTEM SOFTWARE







AT YOUR COMPUTER RETAILER NOW

TRS-80, Apple II, Sorcerer, SOL and Southwest Tech 6800 owners: get more out of your personal computer.

Get into action with G2 Bullseye!, Sea Battle, Confrontation and Attack!

Sharpen your analytical abilities with G2's Outwit I, Outwit II and Mind Bender.

Take a chance with G2's Beat the House. Check out your health with Clinic.

And enjoy the challenging experience of two new G2 computer simulations: The Market and Wildcatting.

Or get serious with three powerful new languages. Level III Basic for the TRS-80. Extended Basic for the SOL. And Standard Basic for the Southwest Tech 6800. All written by Microsoft—the Basic wizards. Exclusively from G2.

Our software has more so your computer does more. Great programming. Highest quality cassette. The most comprehensive instruction manual available. Plus source listing print-out of every application program in Basic. You can learn how the programs were written. We even encourage you to do your own re-programming to improve your skills!

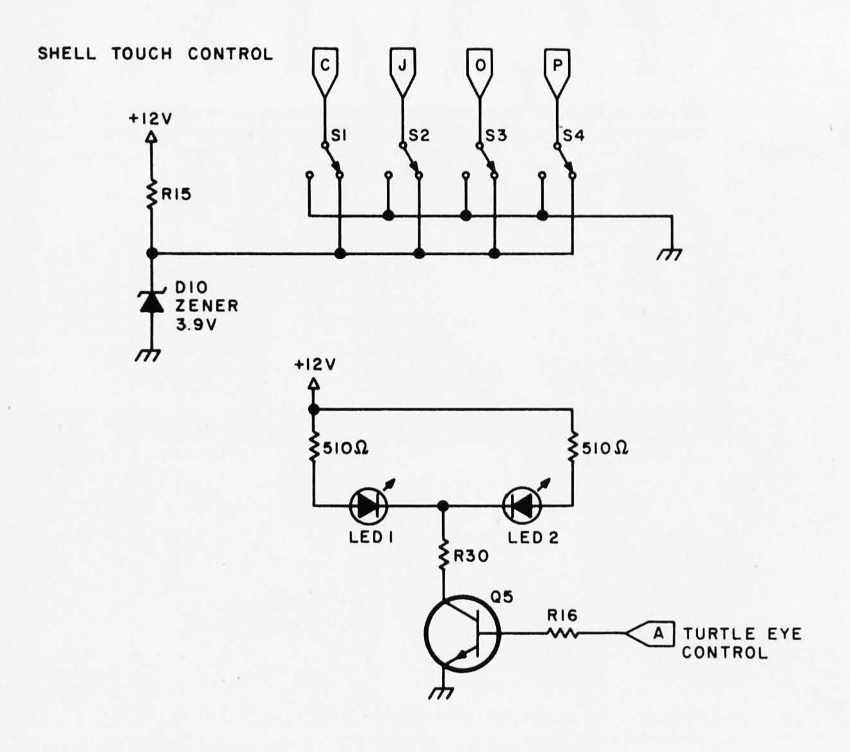
G2 software is available from computer retailers nationwide. If your local retailer doesn't have it, ask him to become a G2 dealer by calling us toll free: (800) 538-8540 (U.S.A.) or (800) 672-8691 (California).

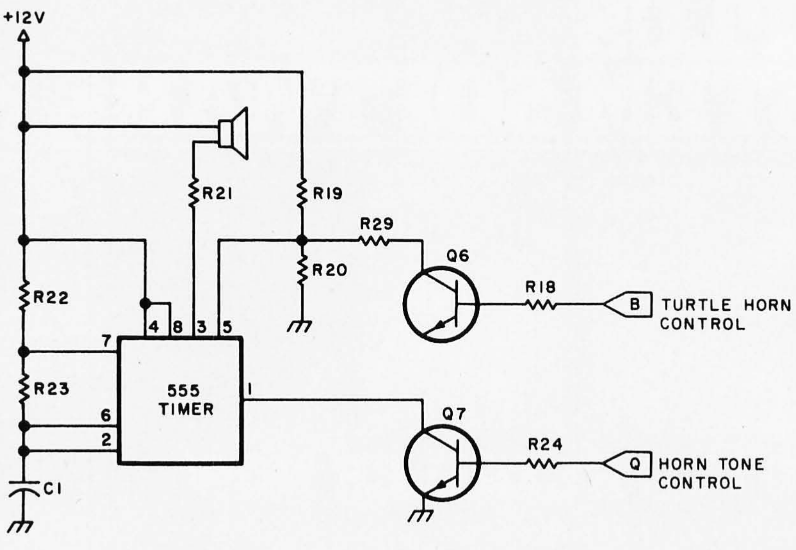
THE REASON YOU BOUGHT YOUR COMPUTER.



GRT CORPORATION
A Product of GRT Corporation
Consumer Computer Group
1286 North Lawrence Station Road, Sunnyvale, California 94086, 408/734-2910

Circle 151 on inquiry card.





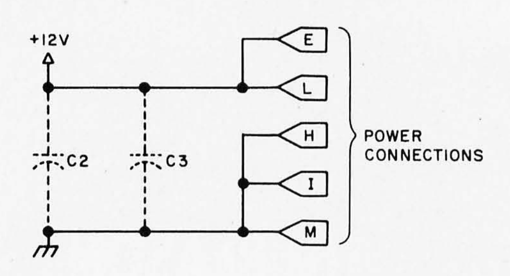


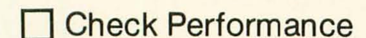
Figure 3: Schematic of the Turtle's touch sensor, lights, and horn control circuitry.

The Anadex 80 Column Printer.



First check the specs. Then check the price.

Our new 80-column dot matrix line printer – the DP-8000 – combines high performance and operating convenience with a low price that's worth checking into.



☐ **Check Performance**

The DP-8000 features a precision engineered, heavy duty printing mechanism that can print the complete 96 ASCII character set, bi-directionally, at 84 LPM.

☐ **Check the Interface**

Included at no extra cost, are two input connectors (see photo) that provide three basic ASCII compatible interfaces: EIA Std. RS232C, for interfacing at up to 96000 Baud with most minicomputers and modems; the 20/60 ma current drive mode required by Teletype® ASR33-35 printers; and the parallel-bit, serial character synchronous Centronics compatible interface.

The DP-8000 includes 12 lines of internal FIFO buffer storage and can accept data continuously or in bursts. Optionally, increased buffer storage of 2048 characters can be supplied for CRT dump and similar applications.

☐ **Check Printer Quality**

A 9 × 7 character font provides virtually half-dot resolution for clean crisp print quality on the original plus three copies.

Precise paper positioning is ensured by a sprocket-feed paper advance, user-programmable Top of Form control, and up to 8 vertical tab positions.

☐ **Check Convenience**

For operating ease, the DP-8000 accepts paper through the rear or bottom of the unit, provides programmable Skip Over Perforation control, and Out of Paper indication and logic signal.

☐ **Check the low Price**

The best news is the price. A complete DP-8000 is unit-priced at under $1000, with substantial discounts in larger quantities.

Once you've checked out the performance and price, we think you'll agree that the DP-8000 is definitely worth checking into.



ANADEX, INC. • 9825 DeSoto Avenue • Chatsworth, CA 91311, U.S.A. • Tel: 213/998-8010 • TWX 910-494-2761
ANADEX, LTD. • Dorna House, Guildford Road • West End, Woking, Surrey GU24 9PW, England • Tel: Chobham (09905) 6333 • Telex: 858762 ANADEX G

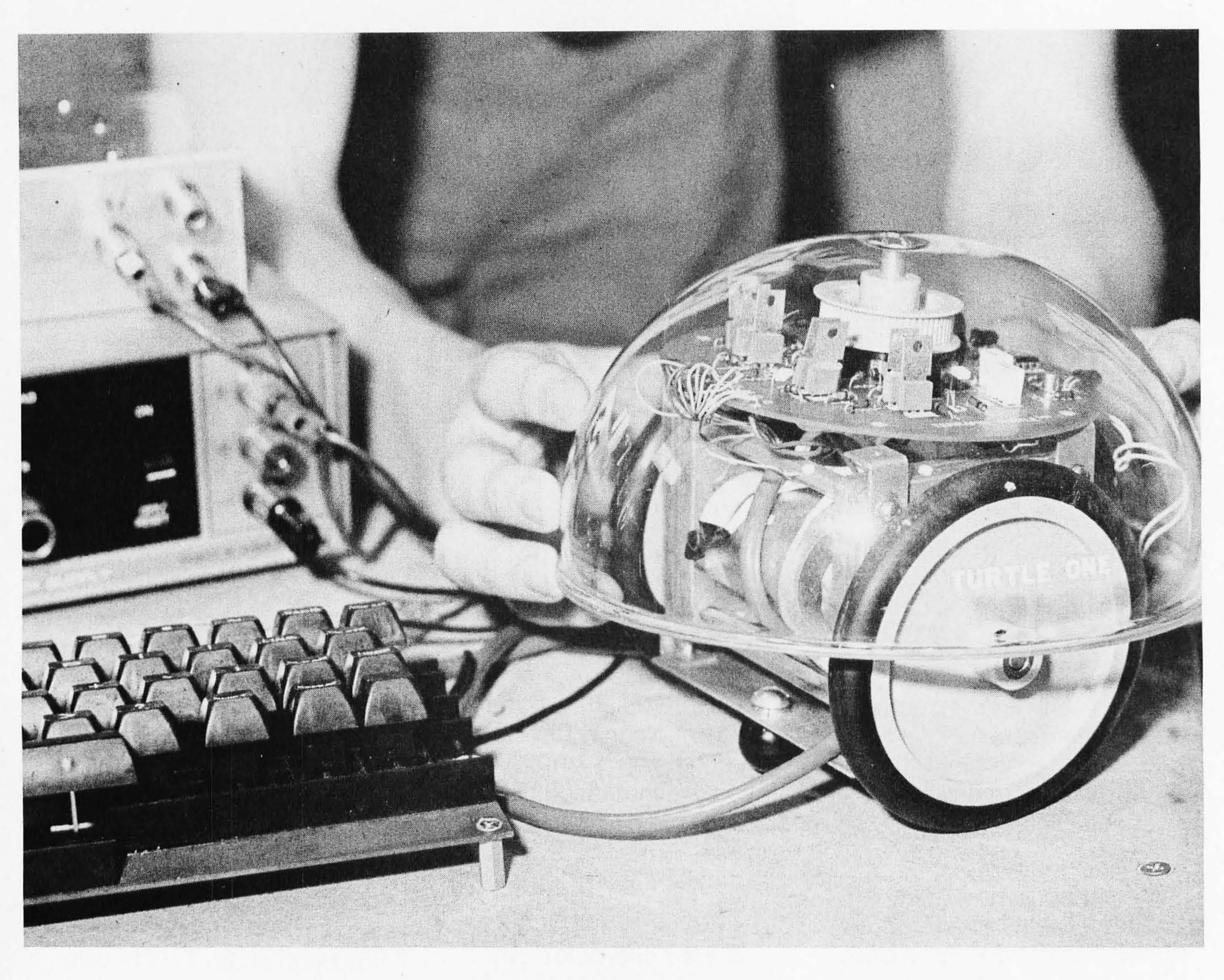


Photo 9: The Turtle with shell attached as a final assembly step.

Text continued from page 79:
sistor driver. Photo 9 shows the attachment of the plastic Turtle shell.

Does the Turtle work? Yes it does, even with a makeshift computer keyboard temporarily substituting for the parallel interface of our computer. The students studied the keyboard's ASCII code and developed a list of keys necessary to command the Turtle's movements, lights, and horn. The Turtle will go under full computer control as soon as an expansion interface can be acquired for our TRS-80 microcomputer.

Those wishing to investigate the Turtle kit, its capabilities, and its cost may obtain full details by writing to:

Terrapin Inc
33 Edinborough St Sixth Floor
Boston MA 02111
Attention: David L McClees, President

In addition, the following address is furnished for those wishing additional information on the application of robots:

United States Robotics Society
Box 26484
Albuquerque NM 87125
Attention: Glenn R Norris, President■

Color. VP-590 add-on Color Board allows program control of 8 brilliant colors for graphics, color games. Plus 4 selectable background colors. Includes sockets for 2 auxiliary keypads (VP-580). $69*

Sound. VP-595 Simple Sound Board provides 256 tone frequencies. Great for supplementing graphics with sound effects or music. Set tone and duration with easy instructions. $24*

Music. VP-550 Super Sound Board turns your VIP into a music synthesizer. 2 sound channels. Program control of frequency, time and amplitude envelope (voice) independently in each channel. Program directly from sheet music! Sync provision for controlling multiple VIPs, multitrack recording or other synthesizers. $49*

Memory. VP-570 RAM Expansion Board adds 4K bytes of memory. Jumper locates RAM in any 4K block of up to 32K of memory. On-board memory protect switch. $95*

EPROM Programmer. VP-565 EPROM Programmer Board comes complete with software to program, copy and verify 5-volt 2716 EPROMs—comparable to units costing much more than the VP-565 and VIP put together! Programming voltages generated on board. ZIF PROM socket included. $99*

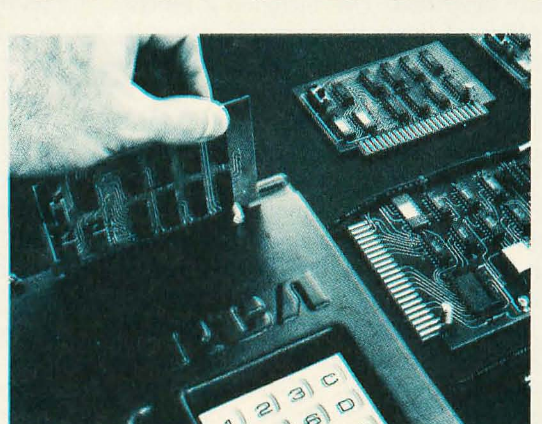
EPROM Interface. VP-560 EPROM Interface Board locates two 5-volt 2716 EPROMs (4K bytes total) anywhere in 32K of memory. VIP RAM can be re-allocated. $34*

ASCII Keyboard.** Fully encoded, 128-character ASCII encoded alpha-numeric keyboard. 58 light touch keys including 2 user defined keys! Selectable upper and lower case. Handsomely styled. Under $50*

Tiny BASIC.** VP-700 Expanded Tiny BASIC Board puts this high-level language on your VIP. BASIC stored in 4K of ROM. Ready for immediate use—no loading necessary. This expanded BASIC includes the standard Tiny BASIC commands plus 12 additional—including color and sound control! Requires external ASCII encoded alpha-numeric keyboard. $39*

Auxiliary Keypads. Program your VIP for 2-player interaction games! 16-key keypad VP-580 with cable ($15*) connects to sockets provided on VP-590 Color Board or VP 585 Keyboard Interface Card ($10*).

COSMAC VIP lets you add computer power a board at a time.



With these new easy-to-buy options, the versatile RCA COSMAC VIP (CDP18S711) means even more excitement. More challenges in graphics, games and control functions. For everyone, from youngster to serious hobbyist. And the basic VIP computer system starts at just $249* assembled and ready to operate.

Simple but powerful—not just a toy.

Built around an RCA COSMAC microprocessor, the VIP includes 2K of RAM. ROM monitor. Audio tone with a built-in speaker. Plus 8-bit input and 8-bit output port to interface relays, sensors or other peripherals. It's easy to program and operate. Powerful CHIP-8 interpretive language gets you into programming the first evening. Complete documentation provided.

Take the first step now.

Check your local computer store or electronics parts house. Or contact RCA VIP Marketing, New Holland Avenue, Lancaster, PA 17604. Phone (717) 291-5848.

*Suggested retail price. CDP18S711 does not include video monitor or cassette recorder.
**Available 1st Quarter, 1979.

See the RCA VIP at the 3rd Annual National Small Computer Show at the New York Coliseum, August 23-26, booth 4211.

The fun way into computers.



Circle 322 on inquiry card.

Few people have ever had the experience of attempting to trace their way through a real maze, so I'm going to ask you to settle back and allow your imagination plenty of freedom.

The absolute darkness surrounding you is barely broken by the candle you carry in your hand. You see enough to know that you are in a featureless corridor, but how far it extends, you can only guess. Somewhere within this maze is a massive gold plate. If you can find this plate, it will become yours and you will be removed to safety.

You are allowed to leave any kind of signs you wish to mark your trail. You know that any corridor you are in will eventually come to a dead end, but it may have any number of similar corridors branching off it. The overall dimensions of the maze are such that the average person could explore it in its entirety before becoming exhausted, if he or she didn't waste a lot of time and energy going in circles.

The Beginning

Years ago I read a brief article about a mechanical mouse that could find its way through simple mazes. Embedded within the walls of the mazes were a number of switches which served as sensors for the brain. The brain was a collection of relays whose points and coils functioned as a large switching network. By trial and error, this mass of hardware could direct the mouse through the maze until it reached the exit.

Over the years, I kept this idea in the back of my mind. I was interested in building such a maze, but the cost and complexity of the project were greater than the potential satisfaction. After I purchased my personal computer, these obstacles disappeared. If I was willing to accept a computer simulation of this project, I could fulfill my dream at no extra expense.

I would need to write a program, of course, but I felt that this would be an easier task than designing and building

My Computer

David E Stanfield
3408 Catalina Dr
Atlanta GA 30341

What follows is the story of how I created a program that would allow my computer to run through mazes similar to the one I've just described. I've included a general description of how the program operates, instructions for using it, and a complete listing of the program.

I regret that I am unable to give you a motion picture of this program in operation. The best I can do is to explain that I first create a maze as simple or complex as I wish on the screen of my video display. I have the cursor operating in its optional nonblinking mode and it therefore appears to be a solid rectangle. As soon as I turn control over to the program, it begins to maneuver the cursor in and out of the various pathways of the maze. The cursor will dodge up and down, back and forth until it eventually finds its goal. The sight of this mad little cursor zooming around the walls of the maze is absolutely fascinating.

Even if you don't intend to get this program up and running, I invite you to come along and explore some areas I found to be quite interesting.

a complete hardware project. As I began to consider what features to include in my program, I came to realize that in several respects, the computer simulation would be superior to the real thing. One important difference was that I could have a maze of greater complexity than would have been feasible with a mechanical version.

Another advantage was the ease with which new mazes could be prepared. Watching repeated runs through the same maze would eventually get monotonous. The choice between shifting and aligning plywood panels or pushing a few keys was no contest. Finally, my entire computer system fits neatly on a small desk. The maze I had visualized making was about the size of my living room.

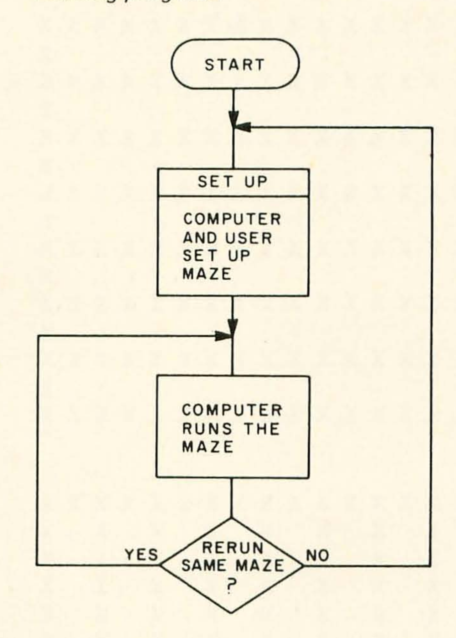
For several weeks I spent much of my spare time considering various ideas for the program. This allowed me the opportunity to explore a wide range of possible features without having to do any actual programming. Gradually, I evolved a straightforward set of goals that I felt would provide an interesting simulation, but would

not be beyond my capability as a programmer.

With these basic goals in mind, I sat down and drew up the simple flowchart you see in figure 1. In graphic form, it indicates that the first function of the program is to assist the user in creating the maze. Once the maze has been prepared, the computer will run the maze until it reaches its goal. At this point, the user can run the same maze again or prepare a new one.

A tremendous number of details had yet to be worked out, but this diagram gave me a secure starting point. Before I could proceed any further I had to make a major decision. From a practical standpoint, designing the program to print out successive sets of coordinates for its moves seemed to be fairly simple. This would mean, however, that the user would have to manually move a marker around on a diagram of the maze. My decision to display the maze on the video display and let the computer move the cursor through the maze increased my work, but made the program far more interesting.

Figure 1: Flowchart of three phases of the maze running program.



Runs Mazes

Program

If you will take a few moments to study figure 2, you will discover that it is really an expanded version of the first block in figure 1. The series of tasks outlined in figure 2 must be performed by the computer and the user in order to set up the maze.

The program begins by having the computer print out a complete list of the commands the user will use to create the maze. After the user indicates his understanding of these commands, the computer issues signals to the display to erase the screen completely. The computer then clears a section of memory that it will later use to remember the maze and the moves through it.

Once the screen and block of memory are cleared, the program prints out a maze and stores a map of it in its special memory. At this point the program allows the user to use a few one letter commands (such as U for up and L for left) to modify the maze. As each command is issued, the program coordinates the making of changes on the screen and the storing of these changes in its block of memory. Once satisfied with the maze modifications, the user issues a final command to signal this fact to the computer. The program will respond by beginning to run the maze.

Maze Creation

The following additional information should help to clarify the above remarks. Once I decided to have the maze displayed on the screen, I needed to select a method of *getting* the maze onto the screen. One method would have been to allow the user to draw the maze on a blank screen. By properly positioning the cursor, the user could have printed a series of Xs anywhere that a wall was desired. I felt that this approach would work, but due to the heavy burden it would place upon the user, I selected another method.

Figure 2: Flowchart expansion of setup block from figure 1.

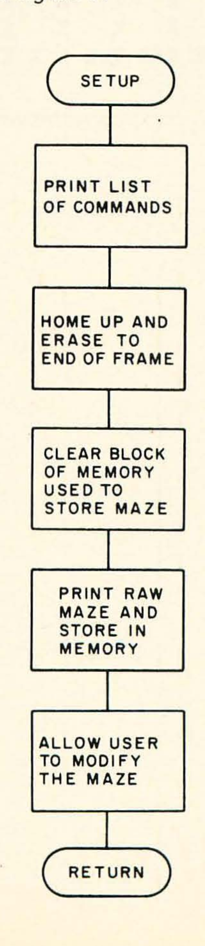


Figure 3: Horizontal paths of maze.

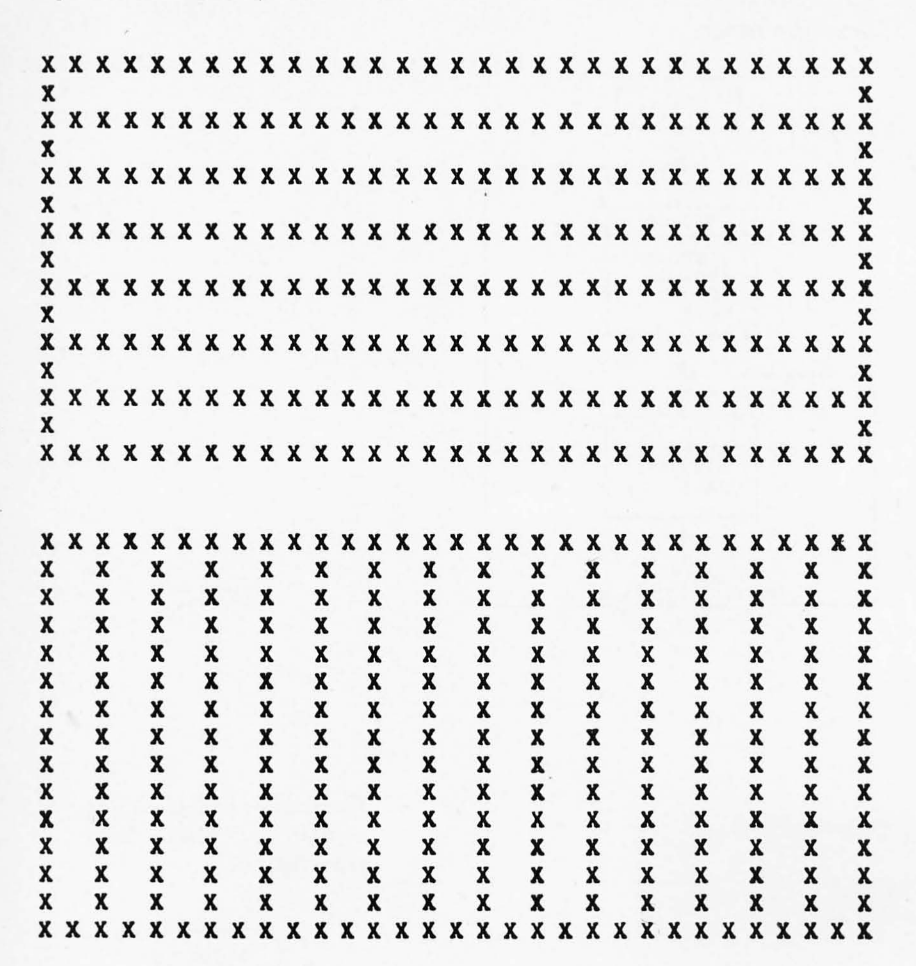


Figure 4: Vertical paths of maze.

Figure 3 shows a box with alternating rows of Xs and spaces. These represent the horizontal paths and walls of the raw maze. Note that there is no way to get from one horizontal path to another. To create such a pathway, the user would simply erase any X on the line of Xs separating them.

Figure 4 represents the vertical paths and walls of the maze. Again, the paths are totally separate, but the user could easily make an opening between the adjacent paths by erasing one X.

Figure 5 is the result of combining figure 3 with figure 4. What you see in figure 5 is the initial maze that I've been referring to. Another way of looking at it is to think of it as an aerial view of a grid of streets running north-south and east-west. Imagine that roadblocks have been established at every intersection. To get from point A to point B, it is necessary only to remove the specific roadblocks blocking your route. For the purposes of this program, the user performs a similar operation by removing those Xs which block the paths he desires through his maze.

This concept is illustrated in figure 6. By erasing the blocking Xs along the desired horizontal and vertical pathways, we are able to create a functional maze. We must remain on those pathways and can erase an X only if it is blocking us unduly. In practice, we can only erase an X if there is a space either above and below it, or to the right and left of it.

The above can be a little confusing, even after you are used to it. Because of this, I built a routine into the program which automatically checks every X you try to erase and determines whether or not that particular X may be erased. If a given X may not be erased, your command will be ignored. As a result, you need not worry about making an error, but remember when you try to erase an X and nothing happens, that the program is designed to do this.

Among the Xs which cannot be erased are those which form the borders of the initial maze, meaning that there is no escape from this maze. It doesn't really matter. Unlike other mazes (in which the idea is to escape), the goal of the computer in this program is to find "food", indicated by the letter F.

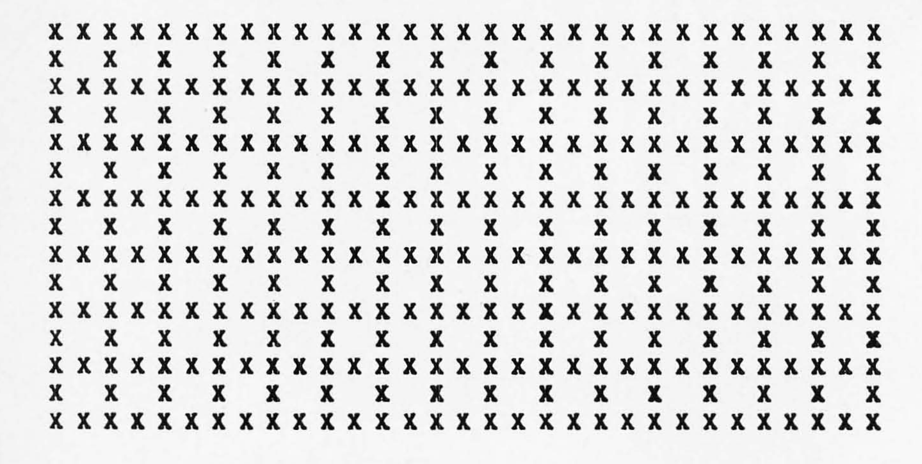
Command Details

While the program does print out a list of all the valid user commands, I feel it is worthwhile to elaborate. One of the basic principles involved in setting up the maze is the fact that the cursor is not allowed to pass beyond any boundary of the maze. This means that when the cursor reaches the right side of the maze, it will not be able to "wrap-around" and reappear on the left side of the screen. It doesn't matter in which direction you are moving: when you reach a boundary line, you will be stopped from going any further. Should you try to go further, the command will be ignored.

I chose this approach to facilitate coordination of the on-screen maze and its counterpart stored in memory. Coordination of the cursor on the screen and the block of reserved memory is critical to the successful operation of this program. It is important that I included a routine designed to abort the program if certain commands are detected.

Because it is so much easier to remember the letter R for right rather than Control-I for right, I decided to allow the user to use U, D, L and R to cause the cursor to move up, down, left and right. In operation, the program recognizes these easy to remember commands and substitutes the specific control character used by the terminal

Figure 5: Complete maze grid, containing all horizontal and vertical components.



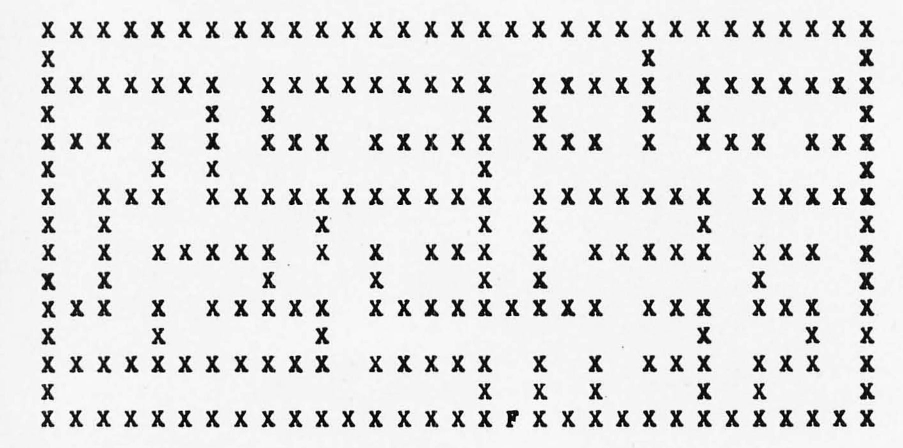


Figure 6: Functional maze produced by removing Xs from the grid of figure 5.

device for the intended cursor movement. I followed this approach because it was relatively easy to program and made things easier for the user. However, should the user input one of the actual control characters recognized by the terminal system, synchronization between the on-screen maze and the special block of memory would probably be lost. Rather than permit this to happen, I chose the simple alternative of having the program check each command input by the user. If it detects any control character, it will immediately erase everything on the screen and return to MIKBUG control. The program itself will not be affected, but you will need to reenter the program at its starting address and begin again.

Other Commands

The remaining valid commands are C, H, F, E, and S. These stand for carriage-return, home-up, food, erase and start. Carriage-return moves the cursor to the left of the current line, while the home-up command is used to move the cursor to the upper lefthand corner of the maze. When you have the cursor positioned directly over an X you wish to remove, use the erase command. Use the food command to replace any X in the raw maze with an F. You may set up several target Fs around the maze, but the program will stop its search as soon as it finds the first one. When you are ready to allow the computer to run the maze you have prepared, issue a start command.

Issuing the start command removes control of events from the user and turns everything over to the computer. As soon as it takes control, the computer checks to see whether the cursor is positioned over an X or a space. At this point, if the cursor is over a space, the computer will begin its search. If, however, the cursor is over an X, the computer will move it to the space in the upper lefthand corner. Either procedure gives the computer a valid point from which to start, and as soon as one has been taken, the computer prepares to make its first move.

Strategy

I now need to discuss, in general terms, the strategy used by the program to move around the maze in search of food. A good place to begin is with figure 7. This type of diagram is referred to as a *tree*. Such a diagram by itself does nothing. When used as an aid in solving certain types of problems, such trees can make it much easier to visualize relationships of various elements of a problem.

If we assume that each circle represents a combination of factors which could be the desired answer, our task is to devise a scheme of searching through the tree until we find the solution. A number of methods for doing this are available. While the one that I selected is not the best, it is reasonably efficient, provides an interesting simulation, and is so simple to understand that I had no reservations about using it as the basis for the most important part of the program.

The strategy I selected for my program is embodied in the following set of simple rules. The first rule is that once a passageway is entered for the first time, it is explored until a dead end is reached. During this initial search, any corridors branching off the passage being explored are ignored. The second rule is that after a dead end is reached, special signals are stored in the reserved block of memory until the computer must either change direction or reach a new corridor. These signals mean that it is not necessary to explore that particular corridor, as it is already known that the goal is elsewhere. In effect, the program will later react to these signs in the same way that

"The data from this experiment could affect national defense. That's why we use Scotch® Brand Disk Cartridges."



Ken Bish, Engineer, Systems Research Laboratories, Inc., Dayton, Ohio

Every Scotch Disk Cartridge is tested and certified error-free before it leaves the factory. Because, for the defense of your data, nothing less than perfection is acceptable.

The disks in Scotch Disk Cartridges are defended by 3M's exclusive CRASHGUARD® protective disk coating. It greatly minimizes the possibility of a head crash, and minimizes the damage, should one occur.

Scotch front-loading Disk Cartridges also feature our exclusive Living Hinge air door, which reduces the possibility of damage to the disk from contact with the air door stop during shipping and handling.

You can get Scotch Disk Cartridges in front or top-loading models. To find out where you can find Scotch Disk Cartridges or virtually any other data recording medium, call toll-free: 800-328-1300. (In Minnesota, call collect: 612-736-9625.) Ask for the Data Recording Products Division.

If it's worth remembering, it's worth Scotch Data Recording Products.



Circle 368 on inquiry card.

it would react to a wall. A third rule is that when the program is retracing previous steps, it will begin exploring new territory as soon as possible.

Search Details

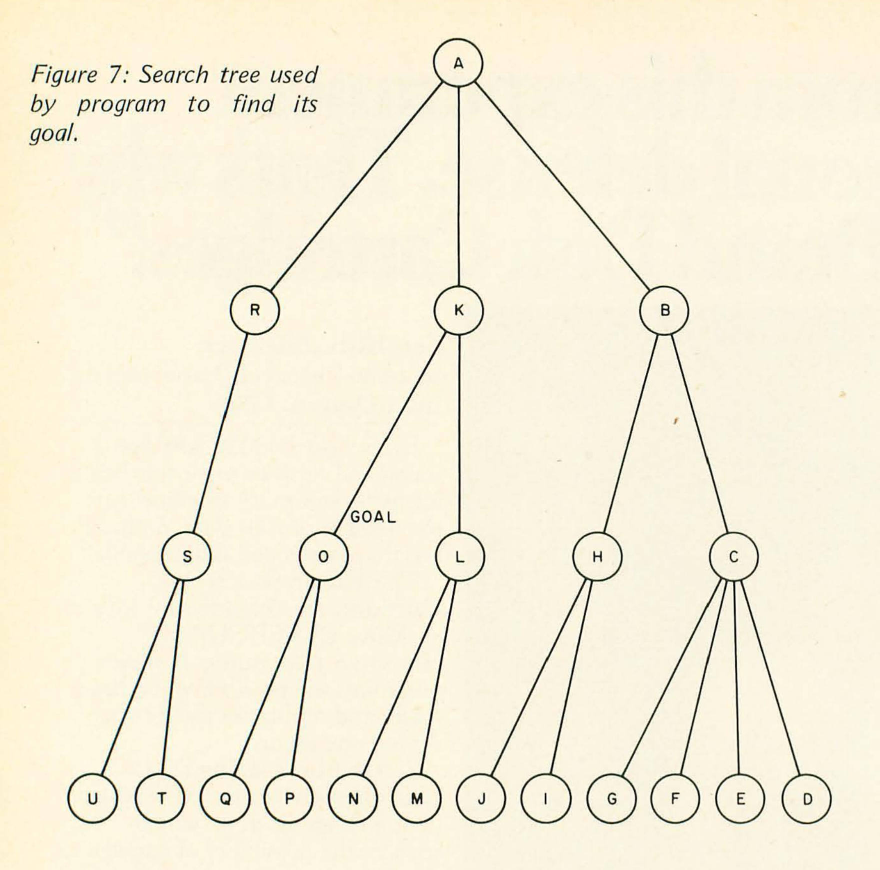
To make these rules clearer, refer to figure 8. Assume that the computer begins its run at position 1. It will move to the right until it reaches the dead end at position 2. Leaving special signals behind itself in memory, the computer will move back to the left until it encounters the new corridor at position 3. As this is an unexplored passage, it will stop retracing its steps toward position 1 and change direction toward position 4. Once it arrives at position 4, it reverses itself and, again leaving the special dead end signs in memory, backs up until it reaches position 5. Here, the computer decides that it has found another unexplored avenue and begins moving toward position 7. Halfway down this corridor it finds the food at position 6 and, having achieved its goal, stops the search.

In addition to the dead end signs, the computer also marks each path it explores with another signal to indicate that it has been there. Both of these signals are stored in the reserved block of memory to serve as guides in choosing the next move. They do not appear on the video display screen (where they would only cause clutter).

If you refer back to figure 7, you will now be able to understand the strategy that I have employed in this program. The procedure is to start at the top and, after arbitrarily choosing one of the branches, descend along it as far as possible. In this instance, we go from A to D. Unable to continue at D, we back up the minimum possible distance to C, where we encounter three unexplored branches. One at a time, we descend from C to E, F, and G. When exploration of these three branches is complete, we have eliminated everything descending from C. Again, we back up the least possible distance. In this case, we move from C back to B. At B we will descend to explore the paths leading to H, I, and J. Once this sequence has been completed, we back up to A. We have, at this point, thoroughly examined one limb of the tree, and use of the same rules over and over will eventually lead us to the goal.

One last point concerning strategy must be covered. Figure 9 illustrates a normal tree with some abnormal additions: closed circular paths, or loops. These may actually be a true representation of a particular problem. They are not, however, used in

Figure 7: Search tree used by program to find its goal.



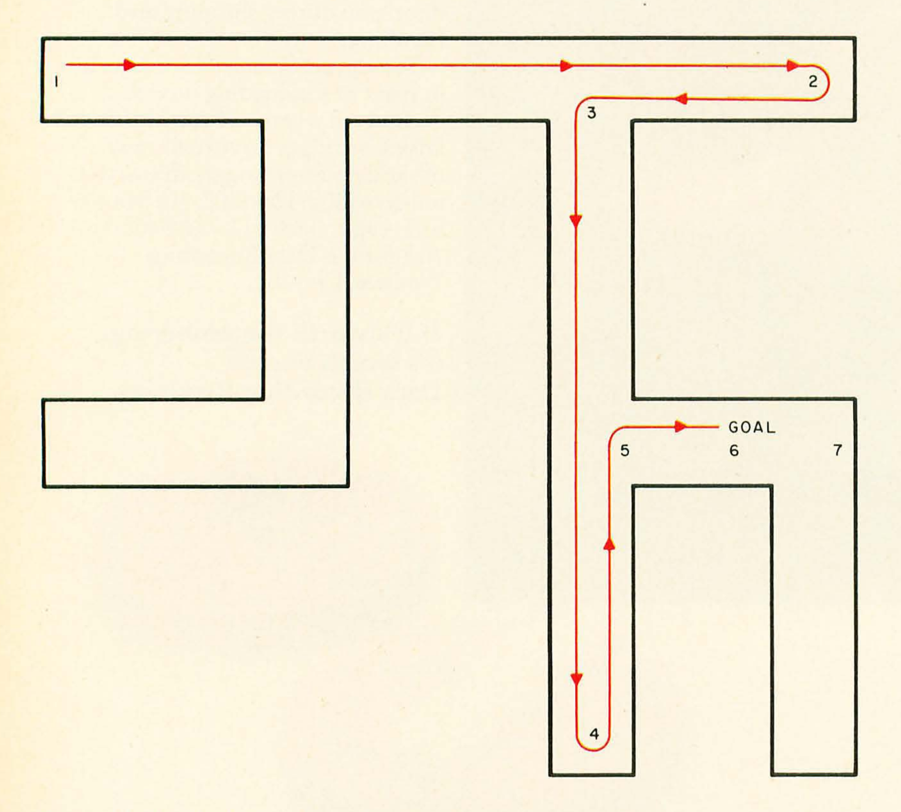


Figure 8: Typical search undertaken by computer. Starting at point 1, the cursor moves right to the dead end at point 2. Marking this position with a special code in memory, the cursor moves left, finding the unexplored corridor at point 3. The cursor changes direction to explore this, until stopped at point 4. Leaving a dead end code, the cursor again reverses, and discovers the new passage at point 5. Changing direction and moving towards point 7, the cursor reaches the "food" at point 6 and stops the search.

classic tree diagrams. Experts in the field of artificial intelligence tend to disapprove of such loops for they can raise havoc with simple search strategies like the one I've been explaining. It is entirely possible to enter one of these loops and, following the rules exactly, remain in the loop indefinitely. This can all be summed up in the following rule: *do not create loops in the mazes you set up for this program to run through.*

Finally, on the matter of loops, figures 10 and 11 are examples of various mazes. Those shown in figure 10 are incorrect because of loops. Those shown in figure 11 are correct.

Run Completion

Now I am going to briefly describe what happens once the computer completes its run. If it was unable to find the goal (because you forgot to include it or placed it out of reach), the program will tell you that it has no valid moves and will ask if you wish to rerun the same maze.

If you indicate that you would like to rerun the same maze, the computer will clean out all the signs it placed in its special block of memory and jump back to that part of the program which allows the user to modify the maze. It will print out a new maze or change the one on the screen. If you desire to start the next run from a new location or further modify the maze, use the same commands you originally used in setting up the maze. When you are ready to begin the new run, input a start command.

When the program actually finds the goal, it goes through the same general routine as when it has no more moves, with one major difference: instead of reporting that it has no more moves, it states that it has found the goal. Beyond this, everything is the same.

If, in either of the above cases, you indicate to the computer that you do not desire to rerun the same maze, the program will erase everything on the screen, completely clear out the special block of memory, and then print out a new initial maze.

Minor Points

A few minor details remain of which you should be aware. Foremost among these is the cursor. I decided to use the cursor to explore the maze because it was the easiest way to do the job. The simulation is very effective when the cursor is operated in the solid mode (as opposed to the normal blinking mode). The program will function perfectly with a blinking cursor, but the visual effect is not as pleasing.

Text continued on page 96

Figure 9: Search tree containing loop paths, which can cause problems.

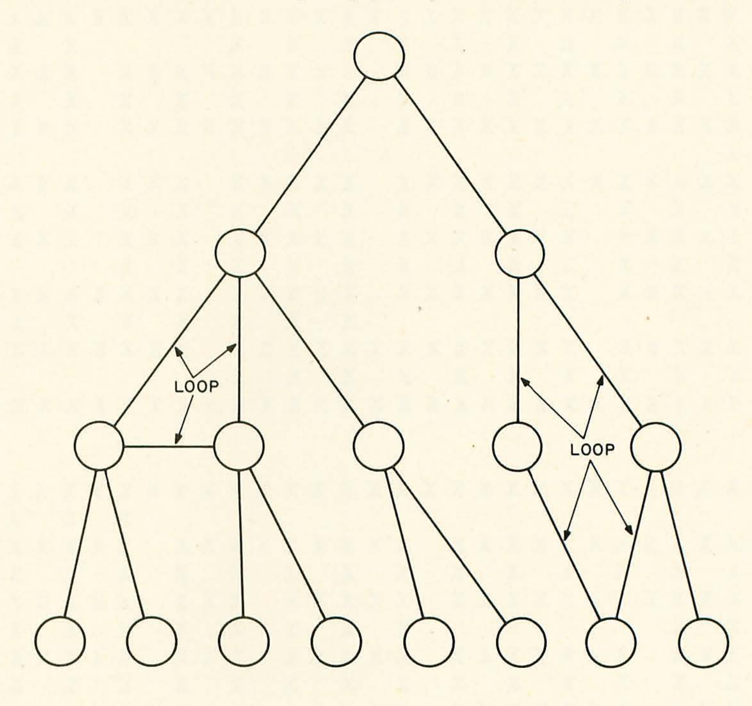
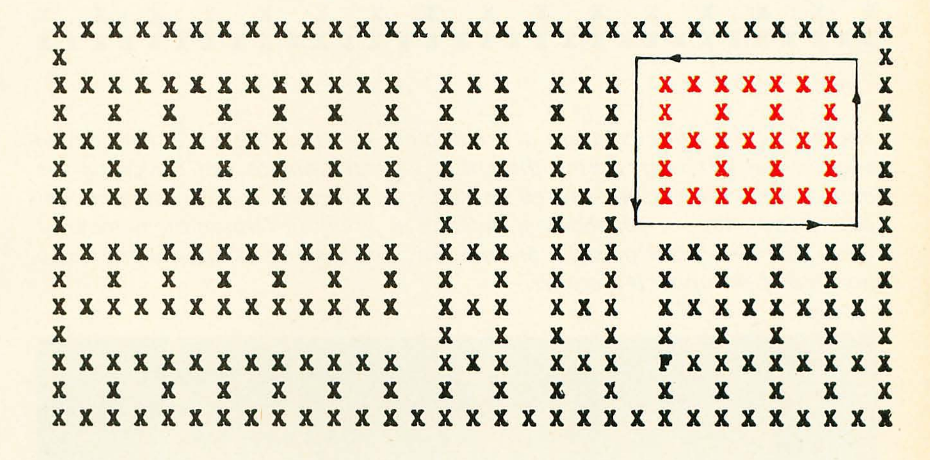


Figure 10: Mazes containing loops. Creation of such mazes is to be avoided.



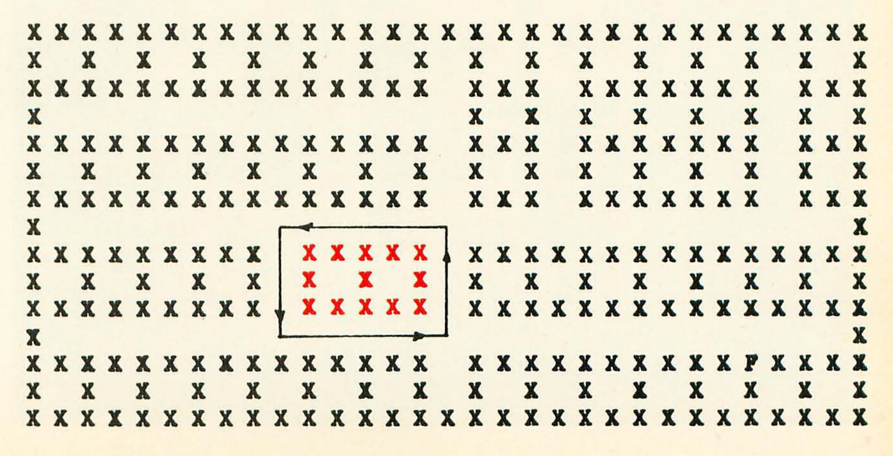
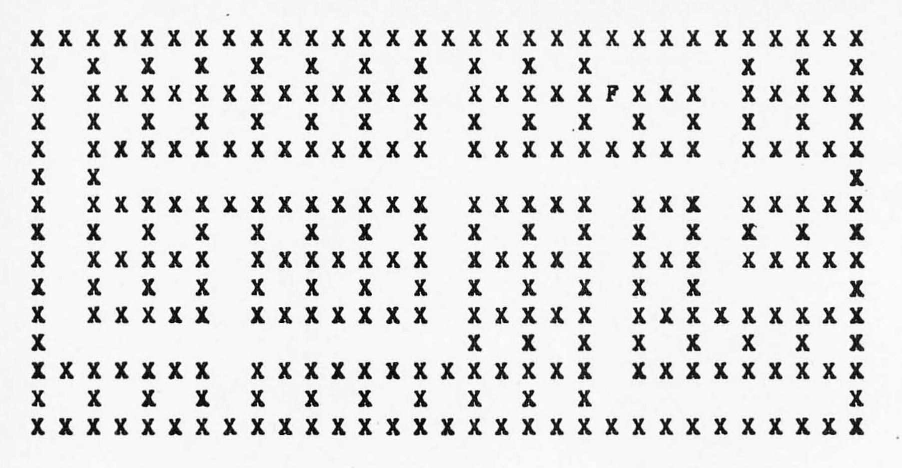


Figure 11: Properly constructed mazes containing no loops.



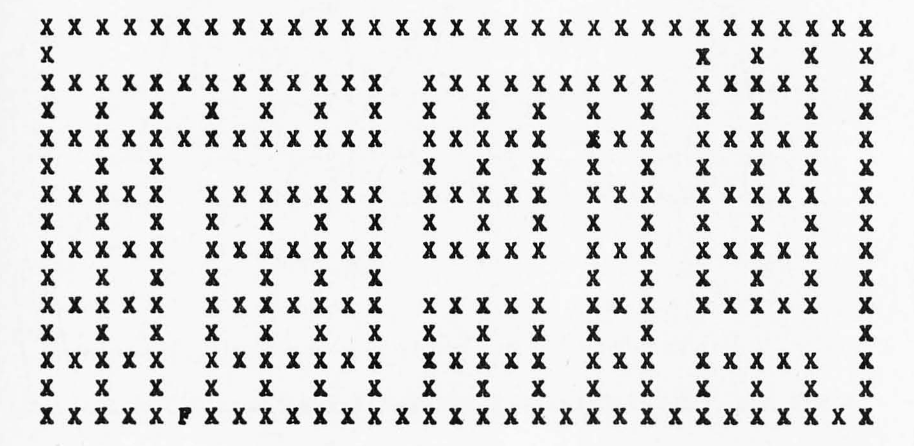
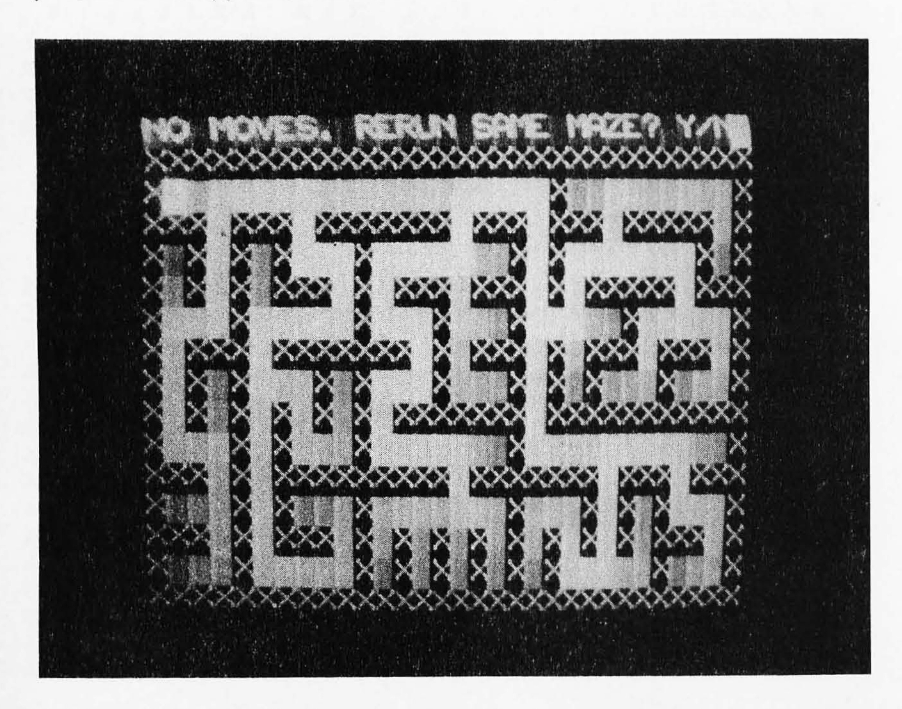


Photo 1: Time lapse exposure of a complete search of the maze. The camera shutter was left open during the entire time the cursor was traversing the maze. When the cursor retraced its steps, it increased the exposure at that point. Thus, the various shadings indicate in a relative fashion the number of times that the cursor passed a given point. The whitest location is the home position at the upper left corner.





Sol.® The small computer that won't fence you in.

A lot of semantic nonsense is being tossed around by some of the makers of so-called "personal" computers. To hear them tell it, an investment of a few hundred dollars will give you a computer to run your small business, do financial planning, analyze data in the engineering or scientific lab — and when day is done play games by the hour.

Well, the game part is true. The rest of the claims should be taken with a grain of salt. Only a few personal computers have the capacity to grow and handle meaningful work in a very real sense. And they don't come for peanuts.

Remember, there's no free lunch.

So before you buy any personal computer, consider Sol.® It costs more at the start but less in the end. It can grow with your ability to use it. Sol is not cheap. But it's not a delusion either.

Sol small computers are at the very top of the microcomputer spectrum. They stand up to the capabilities of mini systems costing four times as much.

No wonder we call it the serious solution to the small computer question.

Sol is the small computer system to do the general ledger and the payroll. Solve engineering and scientific problems. Use it for word processing. Program it for computer aided instruction. Use it anywhere you want versatile computer power!

Build computer power with our software.

At Processor Technology we've tailored a group of high-level languages, an assembler and other packages to suit the wide capabilities of our hardware.

Our exclusive Extended BASIC is a fine example. This BASIC features complete matrix functions. It comes on cassette or in a disk version which has random as well as sequential files.

Processor Technology FORTRAN is similar to FORTRAN IV and has a full set of extensions designed for the "stand alone" computer environment.

Our PILOT is an excellent text oriented language for teachers.

Sold and serviced only by the best dealers.

Sol Systems are sold and serviced by an outstanding group of conveniently located computer stores throughout the U.S. and Canada.

For more information contact your nearest dealer in the adjacent list. Or write Department B, Processor Technology, 7100 Johnson Industrial Drive, Pleasanton, CA 94566. Phone (415) 829-2600.

In sum, all small computers are not created equal and Sol users know it to their everlasting satisfaction.

Circle 300 on inquiry card.

Processor Technology

Author's Note:

The program shown in listing 1 uses several routines from the MIKBUG monitor. Other monitors (SWTBUG, Smartbug) designed to be compatible with MIKBUG should work with the program. However, the program will not work with other types of monitor systems.

Be careful not to type an actual control character while setting up the maze. If you do, you must restart the program at hexadecimal location 003D.

Table 1: Control codes used with the Southwest Technical Products Corporation CT-64 video terminal system.

Control Character	Hexadecimal Code	Function
H	08	Back Space
I	09	Forward Space
J	0A	Line Feed
K	0B	Move Up
P	10	Home Up
U	15	Erase End of Line
V	16	Erase End of Frame
M	0D	Carriage Return

Listing 1: Since the original Maze program was hand assembled over several hand written pages, the machine generated object code shown here in MIKBUG format is the only verifiable representation. Spaces on each line make reading easier, but do not enter the spaces into the computer. The last pair of digits on each line is a checksum. The starting address is hexadecimal 003D.

```
S113 0000 6580 6904 6420 2420 A525 6501 4D00 9580 40
S113 0010 2126 6131 63A4 69AC 2D24 2734 EF94 2D2C 5F
S113 0020 FFCA 27FF 92DF 93FF D9FF 9A5F DAF6 DAFF 60
S113 0030 975E 52FB 99DF DAEB D2EB 5ADB DABD 0100 B3
S113 0040 BD02 00BD 020A BD02 18BD 022C BD03 DCBD 09
S113 0050 02E3 BD02 8CBD 0480 BD04 9ABD 0500 7E00 90
S113 0060 58BD BFDE 09C0 8A5F 92FF 9ACF 9DEA 15D2 C0
S113 0070 9ACA CADB 1AC3 D9D7 12CF 9AFA 9ABB 9157 34
S113 0080 7386 64C0 4635 65A3 6CA5 6724 6537 4401 4F
S113 0090 6131 6127 672D 2C2F 6DB6 6785 2520 27A5 33
S113 00A0 9F5B B24B 8ADA 92D3 C85A BAE8 00DB D8DE 37
S113 00B0 88F9 985B 90EA E0EA D0EE 984A 181B C057 9A
S113 00C0 5136 4085 61F5 653F 6DA4 2537 652C 7EAC BE
S113 00D0 B585 7523 6737 6525 9D25 75B8 2F35 25F9 B1
S113 00E0 8708 D2D2 1FF4 AB59 905D 80D1 19C9 800A 18
S113 00F0 82CA 8A4F 98D8 1859 82C0 1A46 82BD 0200 13
S113 0100 CE01 0ABD E07E BDE1 AC39 1016 434F 4D4D 22
S113 0110 414E 4453 0A0A 0D52 2D4D 4F56 4520 5249 23
S113 0120 4748 540A 0D4C 2D4D 4F56 4520 4C45 4654 D6
```

Listing 1 continued on page 98

Text continued from page 93:

While I was developing the program, I found it desirable to include a delay routine to slow the speed at which the cursor runs through the maze. This delay is used whenever the program is exploring new territory. It greatly increases the impression that the computer is carefully considering each move. The amount of delay can be varied by changing the contents of one location in memory. The specific address is hexadecimal 06BE. You may use any value between 01 and FF. The maximum amount of delay is about one second, which occurs when location 06BE is set to FF. As the value stored in 06BE is decreased, the amount of delay is reduced until, with a value of 01, it is almost unnoticeable.

Hardware Dependence

One final important topic is the configuration of my system. This program is designed around that configuration, and any other could cause problems. Most 6800 system owners should not have any difficulty but, to be on the safe side, I'll go over the details quickly.

First, my terminal system is set up to print 32 characters on each line and 16 lines on each page. When I run this program I operate my terminal in the page mode. Table 1 describes all the control characters used by my system to move the cursor around and to erase the screen. The terminal is set to upper case operations.

Loading the Program

The program (listing 1 is in the MIKBUG tape format) is ready for hand entry. To begin entry, assuming that you are under MIKBUG control, type an L (load) and enter each line exactly as it appears. If you make a mistake, the checksum error detection feature of MIKBUG will catch it and cause the terminal to print a question mark. In order to proceed, again enter an L command and retype the line in question.

Listing 1 continued from page 96:

```
S113 0130 0A0D 552D 4D4F 5645 2055 500A 0D44 2D4D 51
S113 0140 4F56 4520 444F 574E 0A0A 0D45 2D45 5241 FE
S113 0150 5345 2041 4E20 580D 0A43 2D43 4152 5249 E4
S113 0160 4147 4520 5245 5455 524E 0A0D 482D 484F 9B
S113 0170 4D45 2055 500D 0A46 2D52 4550 4C41 4345 9E
S113 0180 2041 4E20 5820 5749 5448 2041 4E20 460D C6
S113 0190 0A53 2D53 5441 5254 0D0A 0A55 5345 2041 D4
S113 01A0 424F 5645 2043 4F4D 4D41 4E44 5320 544F EA
S113 01B0 2050 5245 5041 5245 0D0A 4D41 5A45 2041 67
S113 01C0 4E44 2053 5441 5254 2052 554E 2E20 5748 E9
S113 01D0 454E 2052 4541 4459 0D0A 544F 2050 524F 28
S113 01E0 4345 4544 2C20 5459 5045 2041 4E59 204C F8
S113 01F0 4554 5445 522E 2004 0000 0000 0000 0000 25
S113 0200 CE02 06BD E07E 3910 1604 CE08 008C 0A00 2A
S113 0210 2705 6F00 0820 F639 CE02 28BD E07E CE08 FF
S113 0220 2086 08B7 0227 3900 1016 0A04 BD02 467A 50
S113 0230 0227 7D02 2727 05BD 026B 20F0 CE02 43BD B5
S113 0240 E07E 3910 0A04 861E B702 6A86 58A7 00BD EC
S113 0250 E1D1 7D02 6A27 0608 7A02 6A20 EE08 0886 40
S113 0260 0ABD E1D1 860D BDE1 D139 0086 10B7 026A 1D
S113 0270 8658 A700 BDE1 D108 8620 A700 BDE1 D108 BA
S113 0280 7A02 6A7D 026A 2702 20E6 0139 BD02 E781 0B
S113 0290 5527 2581 4427 2481 4C27 2381 5227 2281 F5
S113 02A0 4827 2181 4327 2081 4527 1F81 4627 1E81 16
S113 02B0 5327 1DBD 02D3 20D4 7E03 8F7E 035C 7E03 AF
S113 02C0 257E 0308 7E02 FC7E 0342 7E04 087E 03C2 70
S113 02D0 7E02 F886 08BD E1D1 A600 BDE1 D186 08BD 45
S113 02E0 E1D1 39CE 0820 39BD E1AC 85E0 2701 39BD 23
S113 02F0 0200 BD02 0A7E E0E3 BD02 D339 BD02 D3BD D4
S113 0300 023C BD02 E37E 028C BD02 D3FF 0323 7C03 C7
S113 0310 24B6 0324 841F 881F 2706 8609 BDE1 D108 5B
S113 0320 7E02 8C00 00BD 02D3 FF03 407A 0341 B603 72
S113 0330 4184 1F88 1F27 0686 08BD E1D1 097E 028C EF
S113 0340 0000 BD02 D3FF 035A B603 5B85 1F27 0309 D0
S113 0350 20F3 860D BDE1 D17E 028C 0000 BD02 D3FF E7
S113 0360 038C B603 8C81 0927 1886 20B7 038E 7D03 7E
S113 0370 8E27 0608 7A03 8E20 F586 0ABD E1D1 7E02 17
S113 0380 8CB6 038D 84E0 88E0 27F4 20DD 0000 00BD F6
S113 0390 02D3 FF03 BFB6 03BF 8108 2718 8620 B703 23
S113 03A0 C17D 03C1 2706 097A 03C1 20F5 860B BDE1 8F
S113 03B0 D17E 028C B603 C084 E088 2027 F420 DD00 BF
S113 03C0 0000 BD02 D3A6 0081 5827 037E 028C 8646 16
S113 03D0 A700 BDE1 D186 08BD E1D1 20EF 86FF B708 B3
S113 03E0 7FB7 08BF B708 FFB7 093F B709 7FB7 09BF 91
S113 03F0 8611 B708 5FB7 089F B708 DFB7 091F B709 A9
S113 0400 5FB7 099F B709 DF39 BD02 D3A6 0081 5827 1A
S113 0410 037E 028C FF04 347C 0435 B604 3584 1F88 C3
S113 0420 1F27 EEFF 0434 7A04 35B6 0435 841F 881F 71
S113 0430 27DF 2002 0000 FF04 34B6 0434 8108 270B B0
S113 0440 B604 3584 E088 E027 C820 09B6 0435 84E0 82
S113 0450 8820 27BD FF04 34A6 0081 1127 0781 FF27 C8
S113 0460 1308 20F3 FE04 3486 20A7 00BD E1D1 BD02 A9
S113 0470 D37E 028C FE04 34B6 0435 8401 27F3 20E7 CE
S113 0480 A600 8120 270C 0101 01CE 0495 BDE0 7ECE 9B
```

I've never seen documentation covering the types of errors the MIKBUG error detection feature will catch, so I purposely made a number of different errors and can report that all were detected. I entered incorrect digits, tried nonhexadecimal characters, rearranged the placement of correct digits and entered the address incorrectly. While I can't guarantee absolute reliability in error detection, I can say that I've entered many programs by hand using this method, and to the best of my knowledge every entry error that I made was caught.

As you successfully enter each line, it is stored in memory. Should you be unable to complete the loading of the entire program in one sitting you may use the tape dump feature of MIKBUG to store what you have entered on cassette or paper tape. At the beginning of your next session, load the tape back into memory and begin hand loading at the point you left off previously.

Eventually you will have the entire program in memory. Before you do anything else, I suggest that you generate a tape of the program. Once that is done, prepare your terminal system to use the program. Set it in the page mode, for upper case operation, with the cursor in a nonblinking mode. The hexadecimal starting address is 003D. Once you have this loaded into addresses A048 and A049, you will be ready to issue the go command.

Because I have entered this entire program by hand on three separate occasions, I can appreciate the feelings of those who think that it's just too big a job. To assist those who are lacking the time or inclination to hand load this program, I will be happy to supply a Kansas City standard cassette tape of the program for a $5 fee.

Conclusion

This program is serious fun. As written, it provides quite a bit of excitement but, human nature being what it is, the urge to improve things may strike one or more of you. Ideas for improvement could include adding land mines, a limited range requiring stops for fuel, magic spots that transport to another location, and even little Klingons. If these or other ideas excite you, I hope you'll get busy and write the program to contain them. I'd like to run it.■

Listing 1 continued:

```
S113 0490 0841 3900 0010 0A0A 0904 A620 8120 2735 E2
S113 04A0 A601 8120 2735 BD04 EAFE 04E7 8120 272F 19
S113 04B0 09A6 0008 8120 272B 017E 067B BDE0 7E7E F5
S113 04C0 0773 1015 4E4F 204D 4F56 4553 2E04 0000 10
S113 04D0 0000 0000 4086 44B7 04D4 3986 5220 F886 D0
S113 04E0 5520 F486 4C20 F000 0000 FF04 E786 20B7 76
S113 04F0 04E9 7D04 E927 0609 7A04 E920 F5A6 0039 10
S113 0500 BD06 CE86 52B1 04D4 2715 864C B104 D427 37
S113 0510 2786 55B1 04D4 2756 8644 B104 D427 3486 9B
S113 0520 09BD E1D1 BD06 BD86 01A7 0008 BD06 CEA6 62
S113 0530 0181 2027 EA7E 0597 8608 BDE1 D1BD 06BD 6D
S113 0540 8601 A700 09BD 06CE 09A6 0008 8120 27E8 78
S113 0550 7E05 9786 0ABD E1D1 BD06 BD86 01A7 00BD 13
S113 0560 058D BD06 CEA6 2081 2027 E87E 0597 860B 43
S113 0570 BDE1 D1BD 06BD 8601 A700 BD04 EABD 06CE 1E
S113 0580 BD04 EAFE 04E7 8120 27E4 7E05 9786 204D 1A
S113 0590 2704 084A 20F9 3939 8652 B104 D427 1B86 26
S113 05A0 4CB1 04D4 2747 8655 B104 D427 6986 44B1 95
S113 05B0 04D4 275F CE04 C27E 04BC BD06 6527 11BD EA
S113 05C0 0660 2713 BD06 5427 15BD 0659 2717 20E4 D6
S113 05D0 BD04 DFBD 0697 39BD 04D5 BD06 AA39 BD04 E7
S113 05E0 DBBD 0681 3986 04A7 00BD 0690 39BD 0665 CA
S113 05F0 27DE BD06 6027 E0BD 0659 2708 BD06 5427 3F
S113 0600 0A7E 05CE BD04 E3BD 068C 3986 04A7 00BD 71
S113 0610 0685 397E 0635 BD06 5427 C3BD 0659 27E4 31
S113 0620 BD06 6527 ABBD 0660 2703 7E05 CE86 04A7 FD
S113 0630 00BD 06AE 39BD 0654 27A4 BD06 5927 C5BD 65
S113 0640 0660 2793 BD06 6527 037E 05CE 8604 A700 B2
S113 0650 BD06 9B39 A601 8101 3909 A600 0881 0139 2B
S113 0660 A620 8101 39FF 0679 8620 4D27 0409 4A20 F6
S113 0670 F9A6 00FE 0679 8101 3900 00BD 0598 7E04 C3
S113 0680 9A86 01A7 0086 09BD E1D1 0839 8601 A700 31
S113 0690 8608 BDE1 D109 3986 01A7 0086 0BBD E1D1 E9
S113 06A0 8620 4D27 0409 4A20 F939 8601 A700 860A C5
S113 06B0 BDE1 D186 204D 2704 084A 20F9 39C6 1086 A9
S113 06C0 FF4D 2703 4A20 FA5A 5D27 0220 F239 FF07 1B
S113 06D0 0086 214D 2704 094A 20F9 8646 A100 2722 D5
S113 06E0 A101 271E A102 271A A120 2716 A122 2712 41
S113 06F0 A140 270E A141 270A A142 2706 FE07 0039 7F
S113 0700 0000 CE07 0BBD E07E 7E07 2C10 1546 4F55 2A
S113 0710 4E44 2049 542E 2052 4552 554E 2053 414D AB
S113 0720 4520 4D41 5A45 3F20 592F 4E04 BDE1 AC81 2F
S113 0730 5927 1381 4E27 0C86 08BD E1D1 8615 BDE1 EA
S113 0740 D120 E97E 077F CE08 20A6 0081 0127 0C81 F5
S113 0750 0427 088C 09FF 2709 0820 EE86 20A7 0020 1B
S113 0760 F286 0DBD E1D1 8615 BDE1 D186 0ABD E1D1 88
S113 0770 7E07 79CE 0716 7E07 058E A042 7E00 4F8E 37
S113 0780 A042 7E00 4043 BF58 97A9 AE19 FF6C 8F51 19
S113 0790 DF55 BF3D 9739 A44B AF39 9A79 8A1B A76C B3
S113 07A0 2092 4806 4887 5802 4106 40A2 40C2 4086 2B
S113 07B0 5897 50C5 60E7 6808 61D0 5086 5C86 68A2 87
S113 07C0 FE3F 9F79 AFBD BFFC FF5D DF75 FF7D DE3D 62
S113 07D0 FF59 9F59 9F7F 8F7F BF6D B731 A671 A77B 4C
S113 07E0 0184 1C82 5082 40C7 10B6 4147 50AB 6806 52
S113 07F0 5086 2041 6182 1882 4186 7003 7003 0083 11
```

Computer Generated Maps,

Part 2

William D Johnston
1808 Pomona Dr
Las Cruces NM 88001

In part 1 (May 1979 BYTE, page 10) we discussed the fundamental techniques involved in the production of computer generated maps and how to apply those techniques to some common map projections. We also presented several simple programs in BASIC which could be easily implemented in your own system to create maps for a variety of purposes. Nevertheless, these simple programs do have their limitations. In this installment we will develop a map projection program which is only slightly more complex, but far more versatile in what it can accomplish.

One of the most interesting projections mentioned in part 1 was the perspective projection, whereby the Earth is shown exactly as it appears to an observer at some specified height above the surface. Several perspective projections were illustrated, but all of these were simplified examples where the observer (or point of projection) was at infinity. It would be much more useful to have a program which would generate maps of the Earth as it appears from *any* chosen altitude and over any desired location. Such a program would give enormous flexibility to displays for space war games and other such practical applications as creating map overlays for weather satellite photographs.

Although details of a program to produce this type of projection were not discussed, the reader was encouraged to investigate the subject of perspective (or projective) geometry to see how the task could be accomplished. By this time many of you have, no doubt, learned that the solution is really quite simple.

Development of Perspective Projection

The key element of the solution can be explained in the following manner: if a line is extended from the center of the Earth to an observer in space, the point on the surface of the Earth that the line passes through is called the observer's *subpoint.* In other words, the observer is directly over this point with respect to the center of the Earth. Now, extend a sight line from the observer to any visible point on the surface of the Earth. You will find that the azimuth angle of the sight line (as measured clockwise from true north) is the same as the great circle bearing from the observer's subpoint to the distant surface point.

Knowing this, you should have been struck by the realization that you might be able to use the same program that was used to generate azimuthal equidistant maps in part 1, to also generate perspective maps. Can it be used? Almost.

The portion of the program which computes the angular component of the polar form of the map coordinates is indeed the same, and the computation of that angle constitutes the major part of the program. As its name implies, however, the azimuthal equidistant projection portrays radial distances uniformly. In a perspective view, distances are not uniform, but become increasingly compressed toward the Earth's limb (ie: edge of visible disk). All we have to do is replace a single statement in the program to correctly compute the radial distance. We will then have the means to produce a perspective projection of the Earth as viewed from any desired altitude over any desired point. By using a simple logic flag, we can choose between either of the two projections and use the same program to generate both types of maps.

You will also recall from part 1 that polar equidistant maps are simply special cases of the azimuthal equidistant map, while the orthographic equatorial and orthographic polar maps are nothing more than special cases of the perspective projection. We can

Listing 1: BASIC subroutine for generating general purpose, perspective projection maps.

```
1000    REM    SUBROUTINE TO COMPUTE MAP COORDINATES
1005    REM    FOR PERSPECTIVE, MODIFIED PERSPECTIVE,
1010    REM    AND AZIMUTHAL EQUIDISTANT PROJECTIONS.
1015    REM
1020    REM
1025    REM    THE FOLLOWING FUNCTION MUST BE DEFINED
1030    REM    BEFORE THIS SUBROUTINE IS CALLED:
1035    REM
1040    REM    FNC( )   COMPUTES THE ARC COSINE OF THE
1045    REM             ARGUMENT.  THE FUNCTION MUST BE
1050    REM             NON-AMBIGUOUS; THAT IS, IT MUST
1055    REM             ATTACH THE CORRECT ALGEBRAIC
1060    REM             SIGN TO THE RESULT.  A GOSUB TO
1065    REM             AN ARC COSINE SUBROUTINE MAY BE
1070    REM             SUBSTITUTED IF DESIRED.
1075    REM
1080    REM
1085    REM    THE FOLLOWING CONSTANTS MUST BE DEFINED
1090    REM    BEFORE THIS SUBROUTINE IS CALLED THE
1095    REM    FIRST TIME, AND ARE NEVER CHANGED:
1100    REM
1105    REM    E  = 6378.0, THE MEAN RADIUS OF THE
1110    REM         EARTH, IN KM.
1115    REM
1120    REM    K1 = 1.0
1125    REM
1130    REM    K2 = 1.5707963, THE VALUE OF PI/2.
1135    REM
1140    REM    K3 = 3.1415927, THE VALUE OF PI.
1145    REM
1150    REM    K6 = 6.2831853, THE VALUE OF 2*PI.
1155    REM
1160    REM    T  = 0.00015, USED AS A TEST VALUE.
1165    REM
1170    REM    Z  = 0.0
1175    REM
1180    REM
1185    REM    THE FOLLOWING VARIABLES MUST BE DEFINED
1190    REM    BEFORE THIS SUBROUTINE IS CALLED THE
1195    REM    FIRST TIME FOR ANY GIVEN MAP.  EACH
1200    REM    TIME A NEW MAP IS TO BE STARTED, THESE
1205    REM    VARIABLES MUST BE REDEFINED BEFORE
1210    REM    CALLING THE SUBROUTINE THE FIRST TIME
1215    REM    FOR THAT MAP, AND ARE NOT CHANGED UNTIL
1220    REM    THE MAP IS FINISHED:
1225    REM
1230    REM    A  IS THE GEOGRAPHIC LATITUDE OF THE
1235    REM       LOCATION ON WHICH THE MAP IS
1240    REM       CENTERED.
1245    REM
1250    REM    A1 IS EQUAL TO SIN(A).
1255    REM
1260    REM    A2 IS EQUAL TO COS(A).
1265    REM
1270    REM    F  IS THE MAP SCALE FACTOR TO CONVERT
1275    REM       TRUE PROJECTED RADIAL DISTANCE TO
1280    REM       THE MAP RADIAL DISTANCE.
1285    REM       F = R/(E*SIN(M)), WHERE R IS THE
1290    REM       RADIUS OF THE FINISHED MAP, IN CM OR
1295    REM       INCHES; E IS AS DEFINED ABOVE; M IS
1300    REM       AS DEFINED BELOW.
1305    REM       F IS USED ONLY FOR PERSPECTIVE AND
1310    REM       MODIFIED PERSPECTIVE PROJECTIONS.
1315    REM
1320    REM    F1 IS THE MAP SCALE FACTOR TO CONVERT
1325    REM       TRUE ARC DISTANCE TO THE MAP RADIAL
1330    REM       DISTANCE.  F1 IS USED FOR AZIMUTHAL
1335    REM       EQUIDISTANT PROJECTIONS ONLY.
1340    REM       F1 = R/M, WHERE R IS THE RADIUS OF
1345    REM       THE FINISHED MAP, IN CM OR INCHES;
1350    REM       AND M IS AS DEFINED BELOW.
1355    REM
1360    REM    G  IS A PRECOMPUTED FACTOR USED IN THE
1365    REM       COMPUTATION OF THE RADIAL COMPONENT
```

Listing 1 continued on page 104

see that by this very simple modification of the azimuthal equidistant map program, we can use a single subroutine to generate any desired perspective map, as well as all of the orthographic, polar equidistant, and azimuthal equidistant maps illustrated in the earlier article! We will add a few frills that will make the program still more versatile.

Modified Perspective Projection

The azimuthal equidistant map portrays radial distances in a linear fashion. On the other hand, radial distances in a perspective map are computed from an involved trigonometric formula in order to show them as they actually appear when viewed from some point in space. Each of these maps has its own special applications, but also has (as do all maps) certain distortions.

A projection which is a compromise between the kinds of distortions inherent in the azimuthal equidistant and the pure perspective maps can be easily developed. This is done by using the same angular component, but presenting the radial distance in direct proportion to the sine of the arc distance. For the sake of simplicity, I will call this a *modified perspective projection.* It has been included as an option in the accompanying program. This projection is quite useful, especially when used in conjunction with the pure perspective projection, as we will see in some later examples. As an added bonus, the sine of the arc distance has to be computed anyway to come up with the angular component of the map coordinates, so we don't have to do any extra work to include this projection in the program.

General Purpose Perspective Projection Program

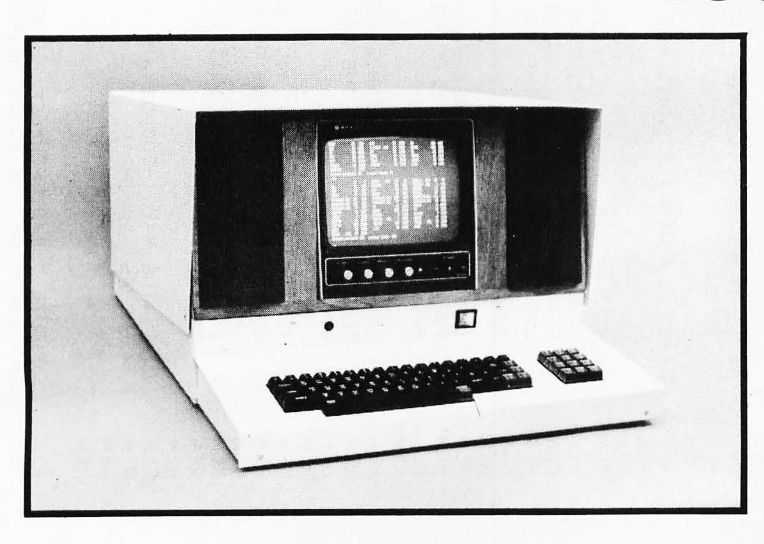
The program in listing 1 is in the form of a subroutine, and is to be used in the same manner as the subroutines presented in part 1. The subroutine is fully documented by the remarks contained within it, so there is little need to elaborate. In fact, of the 300 lines in the listing, fewer than 60 are executable statements; the other 240 or so are all remarks concerning the use and operation of the program.

As in the case of the earlier programs, certain parameters are initialized, then the subroutine is called once for each pair of coordinates to be converted. A geographic latitude and longitude from the data base (see part 1 for a description of the data base) are supplied to the subroutine each time it

Text continued on page 108

Waiting for the right 16 bit computer?

YOUR WAIT IS OVER!



Marinchip Systems M9900 System - 1

■ HARDWARE

16 bit CPU
S-100 bus (11 slots)
60K RAM
Dual 8" diskettes
24 x 80 display

■ SOFTWARE

Network operating system
Extended Commercial BASIC
PASCAL
Word Processor

■ APPLICATIONS

Payroll

Accounts Payable/Receivable

General Ledger

■ If you've been waiting for the right small computer, your wait is over.

■ The Marinchip Systems M9900 System-1 combines a 16 bit processor, powerful system software and the most wanted applications into a ready-to-use desktop computing system—a system that delivers the flexibility, performance and convenience made possible only by the combination of a third generation 16 bit processor and the industry compatible S-100 bus.

■ If you've been waiting for a 16 bit computer, wait no more. The M9900 gives you the TI 9900—not an expanded micro-processor, but a mini-computer on a chip, complete with hardware multiply and divide and multiple register sets. And unlike other 16 bit microprocessors, the 9900 is a member of a complete, compatible, minicomputer family, so if you outgrow the M9900, you don't have to start all over.

■ If you've been waiting for a more powerful operating system, your wait is over. The M9900's Network Operating System is a system without limits—designed to grow as computer hardware evolves, without forcing you to change programs every time you change the hardware. The system provides byte addressable files up to four billion bytes in length, complete dynamic allocation and release, random/sequential access and linked directories for complex file structures. The system automatically pages file data to memory to optimize performance. All I/O devices are treated as files, eliminating device-dependent code in applications.

■ If you've been waiting for a better BASIC, the M9900's Extended Commercial Basic gives you 16 digit accuracy, PRINT USING, random and sequential files, CHAIN with common variables and a choice of interpretive execution for debugging or compilation for production. If you already have applications in BASIC, you'll be glad to know that the M9900's BASIC is similar enough to the most popular 8 bit BASIC that conversion won't be a chore.

■ If you've been waiting for applications you can really use, you'll find our Word Processor, Payroll, Accounts Payable and Receivable and General Ledger to be flexible, easily maintained and ready to use when you plug the system in.

■ If you've been waiting for a computer you can expand when you need to, the M9900's optional 165 CPS line printer ($2500) and 40 Megabyte removable cartridge disc system ($14,000) will make you happy. And since the M9900 uses the S-100 bus, you can choose accessory boards from the thousands available for S-100 systems—we don't tie you to our products.

■ Finally, the M9900 gives you one advantage no other computer can—it's made by Marinchip Systems. We're a computer company, not just a hardware company, software company or applications house. We designed the M9900 from the chips to the applications to bring you a product that advances the state of the art in small computers. Our products are mature, tested items, not dreams, and they're backed up by complete documentation that's been called the best in the industry.

■ Since we're responsible for all the M9900, we're responsive to you. When you have a question, we can give you an answer, not another phone number. If you want to go your own way with the system, we also sell boards and kits that you can put into your own package.

■ The wait for the third generation is over, so don't wait any longer. Call or write today for more detailed information, or order our complete set of manuals.

Complete system $7500

Manuals only $40

Marinchip Systems

16 St. Jude Road
Mill Valley, CA 94941 ■ (415) 383-1545

Circle 206 on inquiry card.

Listing 1 continued from page 102:

```
1370          REM       OF THE POLAR FORM OF THE MAP
1375          REM       COORDINATES FOR PERSPECTIVE
1380          REM       PROJECTIONS ONLY.
1385          REM       G = E * (H2 - E*COS(M)), WHERE E IS
1390          REM       AS DEFINED ABOVE; H2 AND M ARE AS
1395          REM       DEFINED BELOW.
1400          REM
1405          REM    H2 IS THE SUM OF THE EARTH'S RADIUS AND
1410          REM       THE HEIGHT OF THE OBSERVER, AND IS
1415          REM       USED ONLY FOR PERSPECTIVE AND
1420          REM       MODIFIED PERSPECTIVE PROJECTIONS.
1425          REM       H2 = E + H, WHERE E IS AS DEFINED
1430          REM       ABOVE, AND H IS THE HEIGHT OF THE
1435          REM       OBSERVER (POINT OF PROJECTION) ABOVE
1440          REM       THE SURFACE OF THE EARTH, IN KM.
1445          REM
1450          REM    I  IS THE AZIMUTH THAT THE OBSERVER IS
1455          REM       FACING, FROM 0 TO 2*PI RADIANS,
1460          REM       CLOCKWISE FROM TRUE NORTH, AND IS
1465          REM       USED ONLY FOR PERSPECTIVE AND
1470          REM       MODIFIED PERSPECTIVE PROJECTIONS.
1475          REM
1480          REM    J  IS A FLAG WHICH INDICATES THE TYPE
1485          REM       OF PROJECTION TO BE DONE:
1490          REM       J=0, PERSPECTIVE PROJECTION.
1495          REM       J=1, MODIFIED PERSPECTIVE PROJECTION.
1500          REM       J=2, AZIMUTHAL EQUIDISTANT PROJECTION.
1505          REM
1510          REM    L0 IS THE GEOGRAPHIC LONGITUDE OF THE
1515          REM       LOCATION ON WHICH THE MAP IS
1520          REM       CENTERED.
1525          REM
1530          REM    M  IS THE MAXIMUM POSSIBLE ANGULAR
1535          REM       DISTANCE, IN RADIANS, THAT THE
1540          REM       OBSERVER CAN SEE FROM THE POINT OVER
1545          REM       WHICH HE IS LOCATED, TO THE LIMB
1550          REM       (VISIBLE EDGE) OF THE EARTH, FOR A
1555          REM       PERSPECTIVE OR A MODIFIED PERSPECTIVE
1560          REM       PROJECTION.  THIS BECOMES THE
1565          REM       ANGULAR DISTANCE, IN RADIANS, FROM
1570          REM       THE MAP CENTER TO THE EDGE OF THE
1575          REM       MAP.  M = FNC(E/H2), WHERE E AND H2
1576          REM       ARE AS DESCRIBED ABOVE, AND FNC IS
1577          REM       THE ARC COSINE FUNCTION, AS
1578          REM       DESCRIBED ABOVE.
1580          REM       FOR AN AZIMUTHAL EQUIDISTANT
1585          REM       PROJECTION, M IS NORMALLY SET EQUAL
1590          REM       TO PI (WHICH WILL PRODUCE A MAP OF
1595          REM       THE ENTIRE EARTH), BUT THE USER MAY
1600          REM       ALSO SET IT EQUAL TO ANY VALUE LESS
1605          REM       THAN PI IF IT IS DESIRED TO MAP A
1610          REM       SMALLER PORTION OF THE EARTH.
1615          REM
1620          REM
1625          REM    THE FOLLOWING VARIABLES MUST BE DEFINED
1630          REM    PRIOR TO EACH CALL OF THIS SUBROUTINE:
1635          REM
1640          REM    P  IS THE GEOGRAPHIC LATITUDE (FROM THE
1645          REM       DATA BASE) OF THE POINT BEING
1650          REM       PROCESSED.
1655          REM
1660          REM    L  IS THE GEOGRAPHIC LONGITUDE (FROM
1665          REM       THE DATA BASE) OF THE POINT BEING
1670          REM       PROCESSED.
1675          REM
1680          REM
1685          REM    THE FOLLOWING VARIABLES ARE COMPUTED BY
1690          REM    THIS SUBROUTINE:
1695          REM
1700          REM    B1 IS THE ABSOLUTE VALUE OF L1 (SEE
1705          REM       BELOW).
1710          REM
1715          REM    C  IS THE NORMALIZED ANGULAR COMPONENT
1720          REM       OF THE POLAR FORM OF THE MAP
1725          REM       COORDINATES.
1730          REM
1735          REM    C1 IS THE COSINE OF THE VALUE OF C
1740          REM       BEFORE C IS NORMALIZED.
1745          REM
1750          REM    D  IS THE ANGULAR DISTANCE (THE ARC
1755          REM       DISTANCE), IN RADIANS, BETWEEN THE
1760          REM       LOCATION ON WHICH THE MAP IS
1765          REM       CENTERED AND THE POINT BEING
1770          REM       PROCESSED.
1775          REM
1780          REM    D1 IS TEMPORARY STORAGE FOR THE SINE
1785          REM       OF D.
1790          REM
1795          REM    D2 IS TEMPORARY STORAGE FOR THE COSINE
1800          REM       OF D.
1805          REM
1810          REM    L1 IS THE DIFFERENCE BETWEEN THE
1815          REM       LONGITUDE OF THE POINT BEING
1820          REM       PROCESSED, AND THE LONGITUDE OF THE
1825          REM       LOCATION ON WHICH THE MAP IS
1830          REM       CENTERED.
1835          REM
1840          REM    P1 IS TEMPORARY STORAGE FOR THE SINE OF
1845          REM       THE LATITUDE OF THE POINT BEING
1850          REM       PROCESSED.
1855          REM
1860          REM    R1 IS THE RADIAL DISTANCE COMPONENT OF
1865          REM       THE POLAR FORM OF THE MAP
1870          REM       COORDINATES.
1875          REM
1880          REM    S  IS THE OFF-SCALE FLAG.  S=0 MEANS
1885          REM       ON-SCALE.  S=1 MEANS OFF-SCALE.
1890          REM
1895          REM    X  IS THE MAP X-COORDINATE, IN CM OR
1900          REM       INCHES.
1905          REM
1910          REM    Y  IS THE MAP Y-COORDINATE, IN CM OR
1915          REM       INCHES
1920          REM
1925          REM
2000 LET S = Z
2100          REM    COMPUTE THE DIFFERENCE BETWEEN THE
2200          REM    LONGITUDE OF THE POINT BEING PROCESSED
```

Listing 1 continued on page 106

Of course our Floppys aren't cheap.

Quality never is.

It's no use trying to hide the fact: Maxell Floppy Disks give you the finest quality you can buy no matter how much you spend. And more and more of you are agreeing that your data is worth our perfection.

Our floppys work better with your drive.

Naturally, our floppys conform to ISO and IBM specifications. More important, they have also been approved by major OEMs, the people who recommend only those few floppys they are certain will work best with their hardware. So although we obviously don't know which drive system you are using, it makes no difference. Maxell Floppy Disks are so good they actually work better with *any* drive.

What it all means for you.

Yes, you can pay less for some other floppys. But lost data is a terrible price to pay when quality is what you want. And Maxell Floppy Disks help you profit in the very parameters you use this medium for: storing more data with virtually no down-time.

The level of modulation uniformity in every Maxell floppy is vital to double density recording and readout. It means no peak shift, complete freedom from dropouts, total absence of particle orientation. Plus longer life, greater overall durability, and significantly less oxide build-up and head abrasion.

So when you have to depend on full data retrieval, a few cents can make a big difference. Depend on Maxell Floppy Disks. They can really save you.

Maxell offers the full range of Floppy Disks from standard 8-inch to 5¼-inch plus Data Cassettes. *Dealer inquiries invited.*



maxell®
DATA PRODUCTS
The Quality Alternative

Maxell Corporation of America, Data Products Group
60 Oxford Drive, Moonachie, NJ 07074 Tel. (201) 440-8020

Circle 207 on inquiry card.

Listing 1 continued from page 104:

```
2300         REM    AND THE LONGITUDE OF THE LOCATION ON
2400         REM    WHICH THE MAP IS CENTERED, THEN
2500         REM    NORMALIZE THAT VALUE BETWEEN -PI AND
2550         REM    +PI.
2600 LET L1 = L - L0
2700 IF  L1 >= -K3  THEN 3000
2800 LET L1 = L1 + K6
2900 GO TO 5400
3000 IF  L1 <= K3  THEN 4000
3100 LET L1 = L1 - K6
3200         REM    IF THE LONGITUDINAL DIFFERENCE (THE
3250         REM    VALUE OF L1) IS WITHIN 0.00015 RADIAN
3300         REM    OF 0 (OR PI), THEN THE DIFFERENCE IS
3350         REM    CONSIDERED TO BE EXACTLY 0 (OR PI).  IN
3400         REM    SUCH CASES, THE GREAT CIRCLE BEARING, C,
3450         REM    CAN HAVE A VALUE OF ONLY 0 OR PI (NOT
3500         REM    NECESSISARILY RESPECTIVELY), AND THE
3550         REM    STATEMENTS FROM HERE THROUGH STATEMENT
3600         REM    (LINE) NUMBER 4215 PROCESS THE DATA FOR
3650         REM    THESE CONDITIONS.
4000 LET B1 = ABS(L1)
4002         REM    IF THE LONGITUDINAL DIFFERENCE IS NOT
4004         REM    WITHIN 0.00015 RADIAN OF 0, GO TO 4050.
4005 IF  B1 > T  THEN 4050
4010 LET D = ABS(A - P)
4015 IF  D > M  THEN 4300
4020 LET D1 = SIN(D)
4025 LET D2 = COS(D)
4030 LET C = Z
4035 IF  P >= A  THEN 5000
4040 LET C = K3
4045 GO TO 5000
4047         REM    IF THE LONGITUDINAL DIFFERENCE IS NOT
4048         REM    WITHIN 0.00015 RADIAN OF PI, GO TO 4250.
4050 IF  ABS(K3 - B1) > T  THEN 4250
4060 LET D = K3 - A - P
4070 IF  D > K3  THEN 4100
4080 LET C = Z
4090 GO TO 4200
4100 LET D = K6 - D
4150 LET C = K3
4200 IF  D > M  THEN 4300
4205 LET D1 = SIN(D)
4210 LET D2 = COS(D)
4215 GO TO 5000
4220         REM    COMPUTE THE ARC DISTANCE, D, AND THE
4225         REM    GREAT CIRCLE BEARING, C, BETWEEN THE
4230         REM    LOCATION ON WHICH THE MAP IS CENTERED
4235         REM    AND THE POINT BEING PROCESSED.
4250 LET P1 = SIN(P)
4255 LET D2 = A1 * P1  +  A2 * COS(P) * COS(L1)
4260 LET D = FNC(D2)
4265         REM    IF THE COMPUTED ARC DISTANCE IS GREATER
4270         REM    THAN THE MAXIMUM POSSIBLE ARC DISTANCE
4275         REM    FOR THIS MAP, THEN THE POINT IS
4280         REM    OFF-SCALE, AND CONTROL RETURNS TO THE
4285         REM    CALLING POINT AFTER SETTING THE
4290         REM    OFF-SCALE FLAG.
4295 IF  D <= M  THEN 4500
4300 LET S = K1
4400 RETURN
4500 LET D1 = SIN(D)
4520 LET C1 = (P1  -  A1 * D2) / (A2 * D1)
4540 IF  C1 < -K1  THEN 4300
4560 IF  C1 > K1  THEN 4300
4580 LET C = FNC(C1)
4600         REM    NORMALIZE THE VALUE OF C, DEPENDING
4620         REM    UPON THE RELATIVE LONGITUDES OF THE
4640         REM    LOCATION ON WHICH THE MAP IS CENTERED
4660         REM    AND THE POINT BEING PROCESSED.
4680 IF  L1 >= Z  THEN 5000
4700 LET C = K6 - C
4720         REM    AT THIS POINT C IS IN THE RANGE FROM
4740         REM    0 TO 2*PI, MEASURED CLOCKWISE FROM TRUE
4760         REM    NORTH.
4780         REM
4800         REM    IF THE VALUE OF I IS NOT 0, ROTATE THE
4900         REM    COORDINATES BY THE AMOUNT OF I.
5000 IF  I = Z  THEN 5500
5050 LET C = C - I
5100 IF  C >= Z  THEN 5500
5150 LET C = K6 + C
5200         REM    REVERSE THE DIRECTION OF MEASUREMENT OF
5250         REM    C AND ROTATE IT BY PI/2.  THEN
5300         REM    NORMALIZE THE RESULT BETWEEN -PI AND
5350         REM    +PI.  C THEN REPRESENTS THE NORMALIZED
5400         REM    FORM OF THE ANGULAR COMPONENT OF THE
5450         REM    POLAR FORM OF THE MAP COORDINATES.
5500 LET C = K2 - C
5600 IF  C >= -K3  THEN 6000
5700 LET C = C + K6
5800         REM    IF A TRUE PERSPECTIVE PROJECTION IS NOT
5900         REM    TO BE USED, GO TO 6500.
6000 IF  J <> Z  THEN 6500
6050         REM    COMPUTE THE RADIAL COMPONENT, R1, OF
6100         REM    THE POLAR FORM OF THE MAP COORDINATES
6150         REM    FOR A TRUE PERSPECTIVE PROJECTION.
6200 LET R1 = F * (G * D1) / (H2  -  E * D2)
6250 GO TO 7000
6300         REM    IF A MODIFIED PERSPECTIVE PROJECTION IS
6400         REM    NOT TO BE USED. GO TO 6700.
6500 IF  J <> K1  THEN 6700
6520         REM    COMPUTE THE RADIAL COMPONENT, R1, OF
6540         REM    THE POLAR FORM OF THE MAP COORDINATES
6560         REM    FOR A MODIFIED PERSPECTIVE PROJECTION.
6580 LET R1 = F * E * D1
6600 GO TO 7000
6620         REM    COMPUTE THE RADIAL COMPONENT, R1, OF
6640         REM    THE POLAR FORM OF THE MAP COORDINATES
6660         REM    FOR AN AZIMUTHAL EQUIDISTANT PROJECTION.
6700 LET R1 = F1 * D
6800         REM    CONVERT THE POLAR MAP COORDINATES TO
6900         REM    RECTANGULAR MAP COORDINATES.
7000 LET X = R1 * COS(C)
7010 LET Y = R1 * SIN(C)
7020 RETURN
7030 END
```

THE SEARCH FOR A SMALL COMPUTER SYSTEM STARTS HERE

It's the 3rd Annual National Small Computer Show,

presenting the state-of-the-art showcase for micro- and mini-systems technology and software. Here you can survey virtually all makes and models of small computers, whether your interest runs to a no-nonsense micro priced in the hundreds of dollars or a powerful mini costing $20,000 or more. They're all here.

The world of small computers is quite large, extending to business and professional offices, scientific research, medicine and bionics, education, the home and hobbyist, therapeutic applications for the handicapped, design and engineering.

A full selection of lectures is presented to provide a grasp of small systems technology, so that you know what to consider when buying a computer or word processor. It's the first step in discovering what a system can really do for you!

NSCS lectures include sessions on system selection, computer languages, word processing functions, artificial intelligence, software applications, and a dozen more topics for people of all interests.

Plan now to attend. There will be about 30,000 square feet of exhibits, and more than 40 hours of lectures from which to choose. Registration fee is only $5.00 per day, including lectures.

Write for our informational brochure from National Small Computer Show, 110 Charlotte Place, Englewood Cliffs, N.J. 07632.



THIRD ANNUAL NATIONAL SMALL COMPUTER SHOW, New York Coliseum, August 23-26, 1979.

Circle 258 on inquiry card.



Figure 1: A view of the Earth as seen from a point over the southeastern Indian Ocean.

Text continued from page 102:
is called. It returns the rectangular (X,Y) map coordinates (in inches or centimeters) corresponding to that point for the selected projection. The units (inches or centimeters) in which the computerist initially specifies the radius of the finished map automatically determine the units of the map coordinates. All constants, such as π and 0, have been given variable names because most BASIC interpreters can operate faster on variables than on numeric constants.

Remember that all angular parameters are in radians. The program uses spherical trigonometry to arrive at the solution, and some tests have been included to prevent the trigonometric functions from "blowing up" when the map center and the distant point both lie on the same meridian.

The trigonometric functions can also blow up if you attempt to generate an azimuthal equidistant map centered on either of the two poles (ie: a polar equidistant map). This can be avoided by simply specifying the latitude of the map center as slightly less than 90° (perhaps 89.99°). The exact maximum value that can be used will depend upon the precision of your trigonometric routines, but, in any case, you won't be able to see the difference on the finished maps. Incidentally, the longitude that you specify for this kind of map will determine its orientation. This capability was not available with the simplified polar equidistant map program presented in part 1.

For all other types of maps, an option (I) has been included in the program to permit the user to specify the azimuth that the observer in space is facing (ie: to specify the orientation of the map). While this option has little value for a printed map (which the user can turn in any direction), it comes in handy on a video display which is simulating the view from a window of a maneuvering spacecraft.

Within the limits of resolution of any map that you produce, you can assume, for perspective projections, that if the observer is beyond 10,000,000 km above the Earth, the distance is infinite. If you wish to generate an orthographic map, simply assume a height of 10,000,000 km for the observer, and there will be no detectable difference between the resulting map and a truly orthographic map.

Grid lines generally enhance the appearance of perspective maps. These can be included by generating the geographic coordinates within loops in your main (driver) program, then calling the subroutine to obtain the map coordinates. Keep the number of generated points down to the minimum required to obtain the desired resolution, as it is not difficult to expend more processor time creating the grid than creating the map.

Having covered the major operational features of the program (additional details are contained in the remarks within the listing), let us now look at some specific examples. All of the maps illustrated here were generated using the subroutine given in listing 1. Each map was created for a specific purpose and should give you some ideas as to the applications of this program to your own system.

Perspective Maps

Figure 1 provides a good example of what this program can do. Here, the point of projection has been placed at infinity, over a point in the southeastern Indian Ocean. This gives us an excellent view of Australasia, as well as Antarctica. You can create a similar view of any part of the Earth by simply providing the coordinates of the central point. Orthographic perspective

Circle 355 on inquiry card.



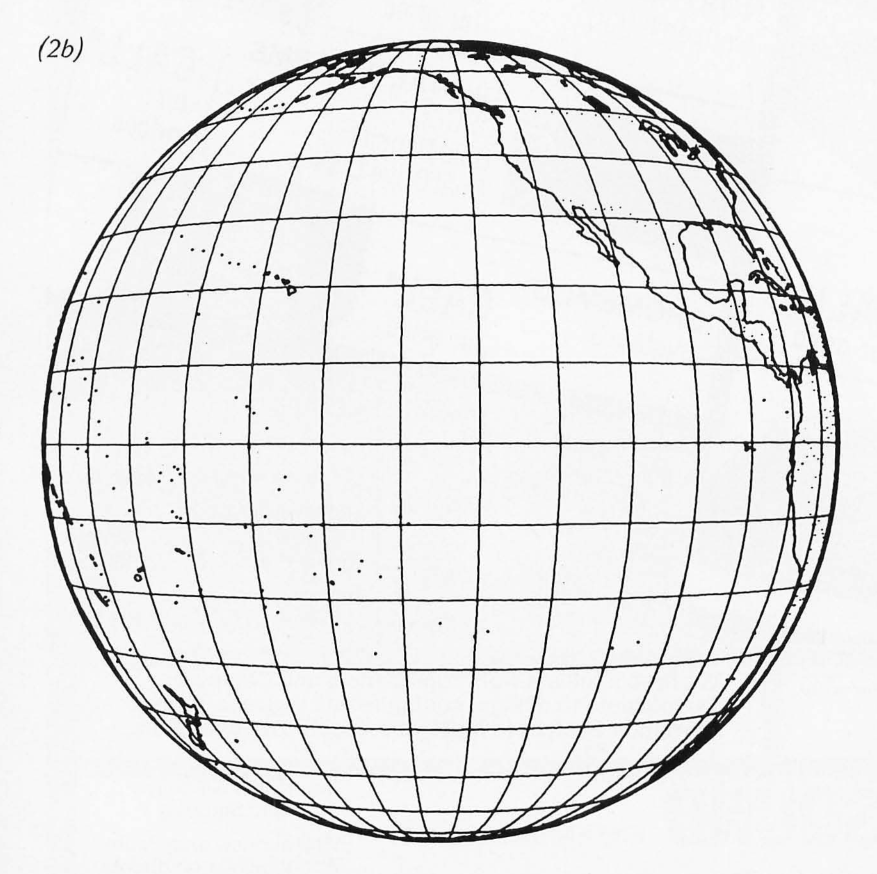


Figure 2: The Earth as seen by a geosynchronous weather satellite. Figure 2a is a view from GOES-2 and figure 2b is from GOES-3. Compare these maps to the weather photographs shown on evening television newscasts.

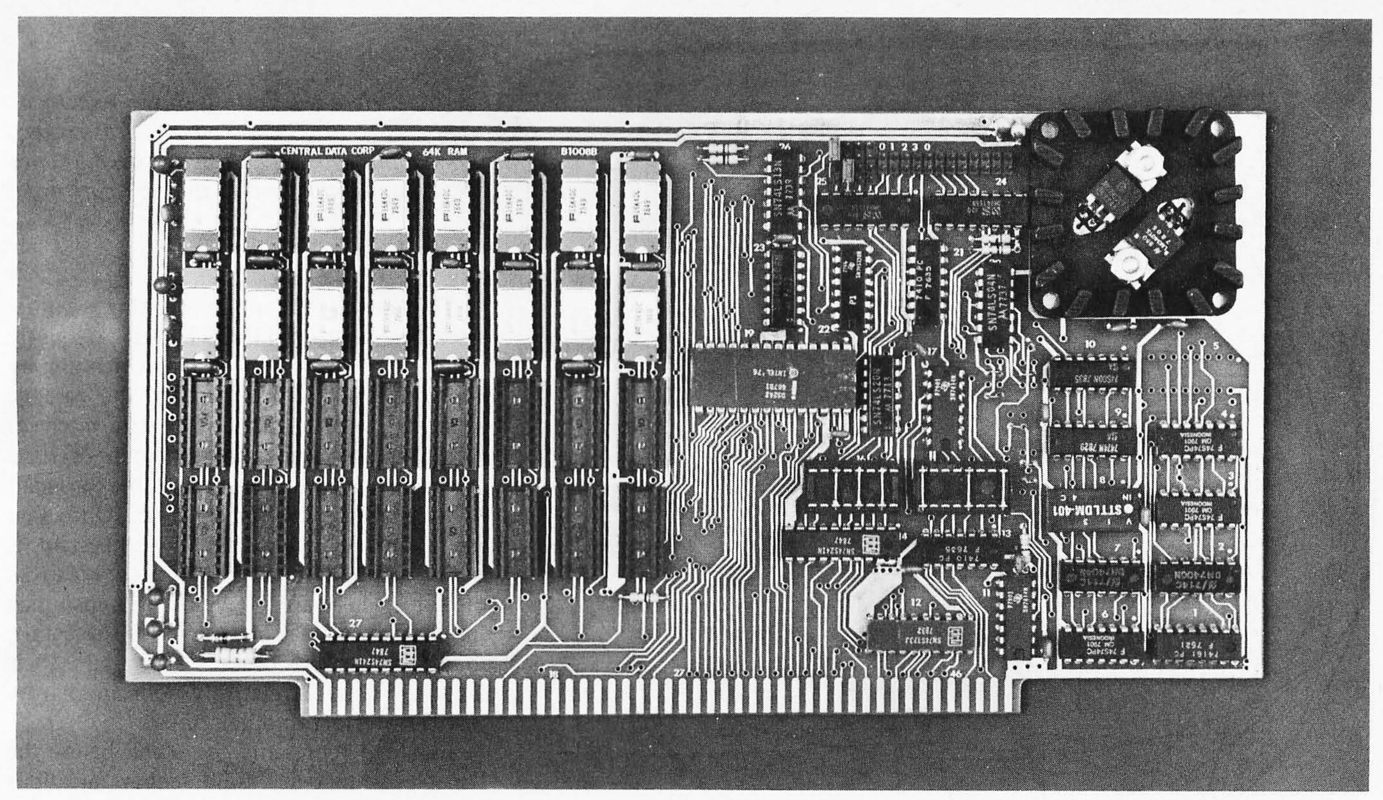
projections, such as this one, find widespread use in scientific applications and are also quite attractive when used in games.

Anyone who watches television weather reports regularly should recognize the maps in figures 2a and 2b. These are the views seen by the two primary United States geosynchronous weather satellites, GOES-2 and GOES-3. (Note that these are not orthographic projections; the field of view extends only about 80° from the central point). Sequences of pictures from these satellites are frequently made into film loops and shown as a sort of jerky motion picture. Although the photographs you see on television usually have outline maps drawn on them, the original pictures transmitted by the satellites do not.

A number of enterprising amateur radio operators and experimenters around the world have built equipment to receive the signals directly from the satellites and print out up-to-the-minute weather pictures in their own homes. Through the use of the perspective projection program, one can generate map overlays in the same scale as the received pictures. This is particularly easy if the weather pictures are being displayed on a video screen where the map can be overlaid electronically.

The same principle is applicable to pictures received from some of the lower altitude polar orbiting weather satellites. Many of the early US APT (automatic picture transmission) satellites, such as those in the ESSA series, used a "snapshot" technique to record the images. The satellite would snap a photograph and transmit it in its entirety before snapping another one. By entering into the computer the altitude of the satellite and the coordinates of its subpoint, one could generate a map overlay to fit the photograph snapped by the satellite at that particular point.

When the next picture was snapped a few minutes later, the satellite would have traveled several hundred miles, but by entering the new coordinates a correct map overlay would be created for each picture. In actual practice, a tracking subroutine is usually incorporated to compute the coordinates of the satellite subpoint. (As exotic as this sounds, it requires only a few simple calculations.)



32K Board Pictured Above

New RAM Prices.

From The Dynamic Memory Company.

16K—$249 32K—$375
48K—$500 64K—$625

Ever since we started making these memory boards over a year ago we have continued to lower our prices to stay competitive. Due to your confidence in us, we are again able to lower our prices! Our reliability has been proven by months of superior performance in thousands of installations. Our low-power boards are being used by quality-minded systems manufacturers across the country and overseas.

4MHz boards now available.
After receiving hundreds of requests, our engineering staff has come up with a new version of our board which runs on 4MHz Z-80 systems. It wasn't easy to come up with a high speed board which would operate as reliably as our 450ns version, but after months of careful design and testing, we did it. The price of the 250ns board is $10 per 16K additional.

All of our features remain.
Our boards didn't become great sellers only because of the price. We still offer you our deselect feature which allows our RAM to overlap with any fixed memory areas in your system. Also, the RAM area of our board is fully socketed so that you can expand the board yourself.

Other standard features include: plug selectable addressing on 16K boundaries (shorting plugs are placed over wire-wrap pins to address the board — located on the top of the board for easy changes), S-100 and Z-80 compatability and totally invisible refresh — no wait states.

Fully assembled, tested, and guaranteed.
All of our boards go through a rigorous testing procedure. They are then placed on burn-in running a series of memory tests to detect any other possible faults. After you receive the board, you are backed by us with a one year warrantee.

Low power consumption keeps your computer from "losing its cool."
The total power consumption of our 16K board is typically less than 4 watts (+8V @ 300ma, +16V @ 150ma and −16V @ 20ma). Boards with additional memory typically increase power consumption only 1 watt per 16K!

Standard S-100 Interface.
Our board is designed to interface with any standard S-100 CPU. All of the timing of the board is independent of the processor chip, and the board is set up for different processors by changing two plugs on the board.

Contact your local dealer.
To find out more about our RAM boards, contact your local dealer. If he is unable to help you, call or write us for a fast response. Central Data Corporation, 1207 North Hagan Street, Champaign, IL 61820. (217) 359-8010

Central Data

Circle 45 on inquiry card.



Figure 3: Here the Earth is shown exactly as it appears to the Oscar 7 amateur radio communications satellite as it passes over Atlanta GA. Compare this map to that in figure 4.

The newer US polar orbiting weather satellites, such as TIROS-N, use a slightly different transmission system which greatly improves the picture quality, the ease of reception, and the amount of data received. It makes the generation of map overlays more difficult, however, and the program presented here cannot be used. (Once you are familiar with the transmission system, it is not difficult to develop a program to do the job.) There are a few satellites, including some of the Soviet Meteor series spacecraft, that still use the older system, but their picture quality is relatively poor and hardly worth the effort to obtain them.

Figure 3 represents the scene below the Oscar 7 amateur radio communications relay satellite as it passes over Atlanta GA. Anyone engaged in satellite communications would do well to have available the capability for such a display. Updated in real time, it provides a continuous panorama of the area visible from the satellite and, hence, the area with which communications through the satellite are possible at any particular time. Any two or more stations can talk to one another as long as they are located within the mapped area. As the satellite moves in its orbit, the mapped area changes, but as long as your own location is within the map you can talk to all other points on the map.

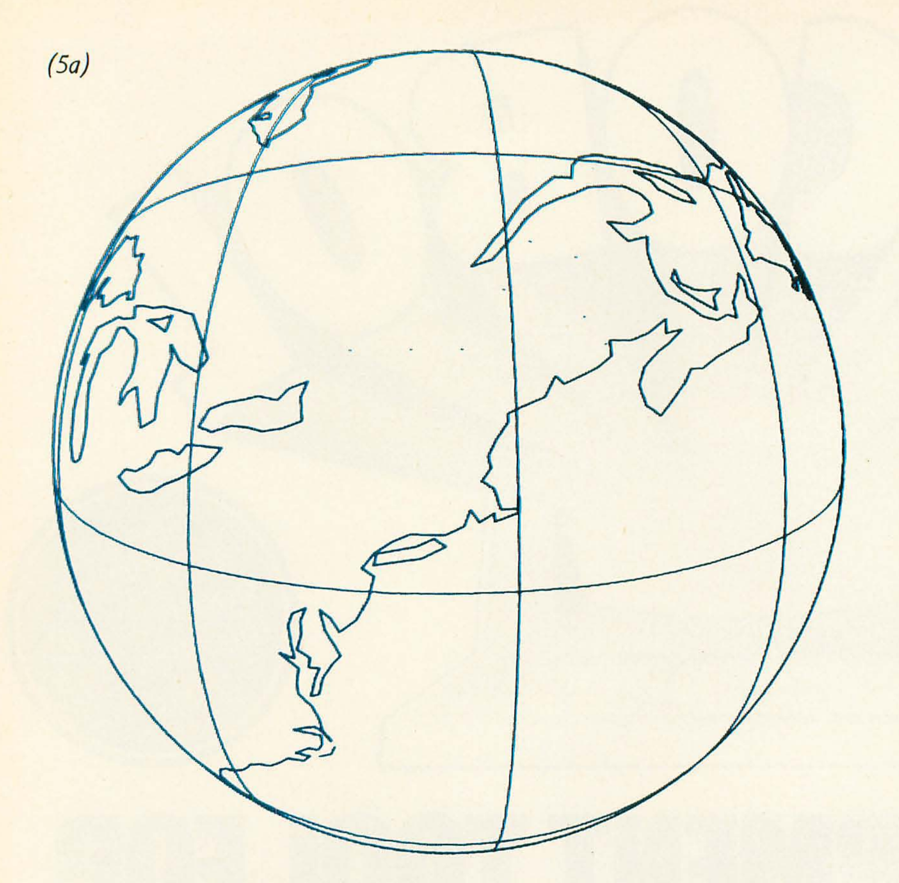
Modified Perspective Maps

We have also come upon the ideal occasion to make use of the modified perspective projection. Figure 4 illustrates the same area as that of figure 3, but the modified projection has been used to reduce some of the distortion inherent in the pure perspective version. Note the differences between the figures, especially in the west coast areas of the US, the northern coast of South America, and the upper reaches



Figure 4: This is the same view as that in figure 3, but here we have used the modified perspective projection to reduce some of the distortion. Compare the west coast of the United States, the northern coast of South America, and the upper reaches of Hudson Bay. Remember that both maps cover the same area.

(5a)



(5b)

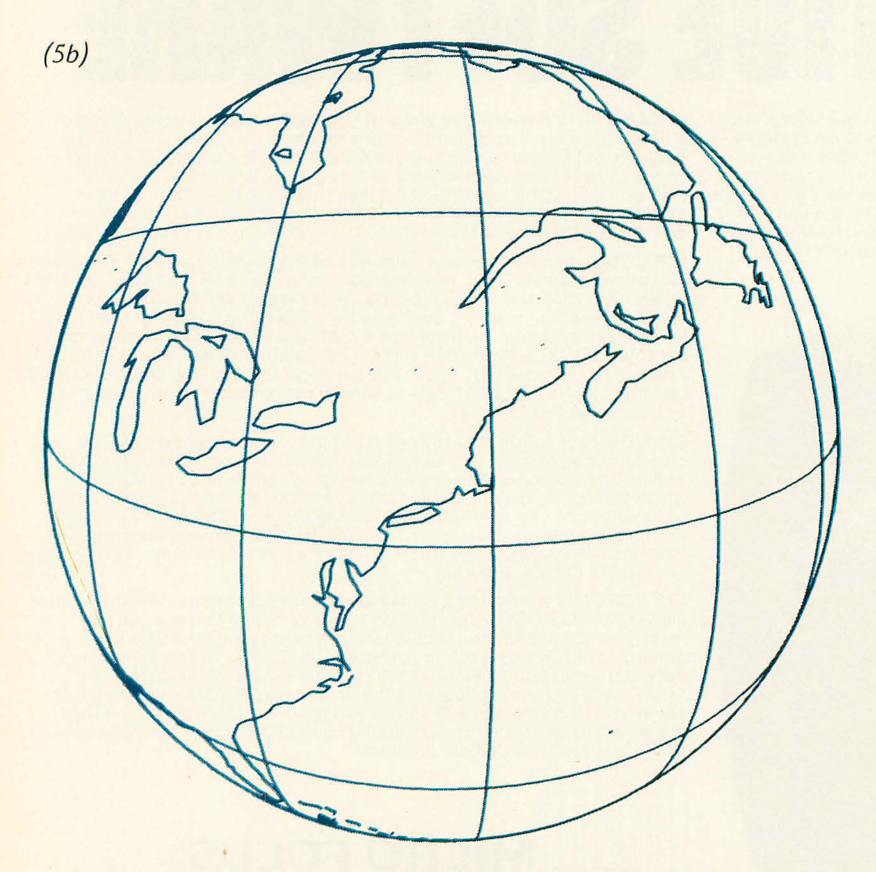


Figure 5: This is a sequence of views of the Earth as one would see it out the window of a spacecraft taking off from Peterborough NH. The final two views are orthographic, with the last of these illustrating the view after the spacecraft has made a 45° turn to the right. (Figures 5c, 5d, 5e and 5f are shown on pages 118 and 119.)

of Hudson Bay. While figure 3 reproduces the scene as it appears visually, figure 4 shows more clearly all of the areas with which communications can be established when the satellite is at the given point.

Incidentally, all of the Oscar satellites are at altitudes comparable to those of the various polar orbiting weather satellites. Indeed, they are launched on the same rockets. The Oscars take the place of otherwise useless ballast and are ejected a few minutes before or after the weather birds. Consequently, the pictures transmitted from this type of weather satellite, especially the earlier versions, are views similar to that shown in figure 3.

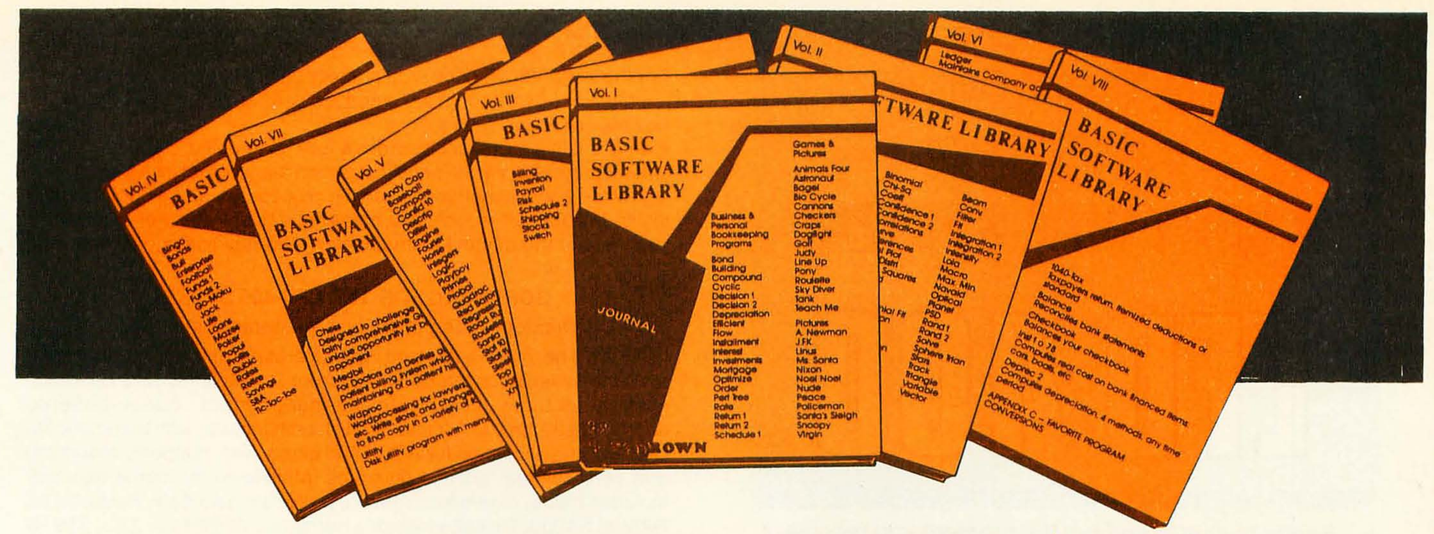
Maps for Space Games

Let us now take a look at some maps that will be of special interest to space game fans. Figures 5a through 5f comprise a sequence of views of the Earth as seen from the window of a spacecraft taking off from Peterborough NH (where BYTE Publications is located). The particular altitudes used in generating these figures were chosen arbitrarily, but they could just as well be input from the game program itself. A fairly large altitude change is required to get a significant change of scenery (assuming no lateral movement). Therefore, it is not necessary to update the display very often if the spacecraft is ascending or descending vertically. Whenever the craft is moving laterally, however, you will want to change the display more often.

The final two views in the sequence of figure 5 are orthographic; that is, the point of projection is at infinity. Although this is not truly realistic in terms of what space travelers see as they recede from the Earth, it is typical of the display that the ship's navigator might have on his video console, regardless of altitude. (Of course, the navigator would want to have a map display of an area considerably larger than what could be seen out the window.)

The last view of the sequence shows the orientation after the spacecraft has made a 45° turn to the right (assuming the observer

Text continued on page 122



COMPUTER SOFTWARE

For Homeowners, Businessmen, Engineers, Hobbyists, Doctors, Lawyers, Men and Women

We have been in business for over nine years building a reputation for providing a quality product at nominal prices — NOT what the traffic will bear. Our software is:

- **Versatile** — as most programs allow for multiple modes of operation.
- **Tutorial** — as each program is self prompting and leads you through the program (most have very detailed instructions contained right in their source code).
- **Comprehensive** — as an example our PSD program not only computes Power Spectral Densities but also includes FFT's, Inverse-transforms, Windowing, Sliding Windows, simultaneous FFT's variable data sizes, etc. and as a last word our software is:
- **Readable** — as all of our programs are reproduced full size for ease in reading.
- **Virtually Machine Independent** — these programs are written in a subset of Dartmouth Basic but are not oriented for any one particular system. Just in case your Basic might not use one of our functions we have included an appendix in Volume V which gives conversion algorithms for 19 different Basic's; that's right, just look it up and make the substitution for your particular version. If you would like to convert your favorite program into Fortran or APL or any other language, the appendix in Volume II will define the statements and their parameters as used in our programs.

Over 85% of our programs in the first five volumes will execute in most 8K Basic's with 16K of free user RAM. If you only have 4K Basic, because of its lack of string functions only about 60% of our programs in Volumes I through V would be useable, however they should execute in only 8K of user RAM.

For those that have specific needs, we can tailor any of our programs for you or we can write one to fit your specific needs.

Vol. I

Business & Personal Bookkeeping Programs

Bond, Building, Compound, Cyclic, Decision 1, Decision 2, Depreciation, Efficient, Flow, Installment, Interest, Investments, Mortgage, Optimize, Order, Pert Tree, Rate, Return 1, Return 2, Schedule 1

Games & Pictures

Animals Four, Astronaut, Bagel, Bio Cycle, Cannons, Checkers, Craps, Dogfight, Golf, Judy, Line Up, Pony, Roulette, Sky Diver, Tank, Teach Me

Pictures

A. Newman, J.F.K., Linus, Ms. Santa, Nixon, Noel Noel, Nude, Peace, Policeman, Santa's Sleigh, Snoopy, Virgin

Vol. II

Binomial, Chi-Sq., Coeff, Confidence 1, Confidence 2, Correlations, Curve, Differences, Dual Plot, Exp-Distri, Least Squares, Paired, Plot, Plotpts, Polynomial Fit, Regression, Stat 1, Stat 2, T-Distribution, Unpaired, Variance 1, Variance 2, XY

APPENDIX A

Beam, Conv, Filter, Fit, Integration 1, Integration 2, Intensity, Lola, Macro, Max. Min., Navaid, Optical, Planet, PSD, Rand 1, Rand 2, Solve, Sphere Trian, Stars, Track, Triangle, Variable, Vector

Vol. III

Billing, Inventory, Payroll, Risk, Schedule 2, Shipping, Stocks, Switch

Vol. IV

Bingo, Bonds, Bull, Enterprise, Football, Funds 1, Funds 2, Go-Moku, Jack, Life, Loans, Mazes, Poker, Popul, Profits, Qubic, Rates, Retire, Savings, SBA, Tic-Tac-Toe

Vol. V

Andy Cap, Baseball, Compare, Confid 10, Descrip, Differ, Engine, Fourier, Horse, Integers, Logic, Playboy, Primes, Probal, Quadrac, Red Baron, Regression 2, Road Runner, Roulette, Santa, Stat 10, Stat 11, Steel, Top, Vary, Xmas

APPENDIX B

Vol. VI

Ledger	Maintains Company accounts and generates financial reports. Includes routines for: Pyrl, Inv, Depr, A/R, A/P.

Vol. VII

Chess	Designed to challenge the average player, fairly comprehensive. Great fun for all, offers a unique opportunity for beginners in need of an opponent.
Medbil	For Doctors and Dentists alike, a complete patient billing system which also permits the maintaining of a patient history record.
Wdproc	Wordprocessing for lawyers, publishers, writers, etc. Write, store, and change from rough draft to final copy in a variety of formats.
Utility	Disk utility program with memory testing.

Vol. VIII

1040-Tax	Taxpayers return, itemized deductions or standard
Balance	Reconciles bank statements
Checkbook	Balances your checkbook
Inst 1 o 78	Computes real cost on bank financed items; cars, boats, etc.
Deprec 2	Computes depreciation, 4 methods, any time period

APPENDIX C — FAVORITE PROGRAM CONVERSIONS

Vol. I — $24.95 Bookkeeping Games Pictures	Vol. II — $24.95 Math/Engineering Plotting/Statistics Basic Statement Def.	Vol. III — $39.95 Advanced Business Billing, Inventory Investments Payroll	Vol. IV — $9.95 General Purpose	Vol. V — $9.95 Experimenter's Program	Vol. VI — $49.95 Mini-Ledger	Vol. VII — $39.95 Professional Programs	Vol. VIII — $19.95 Homeowner's Programs

AVAILABLE AT MOST COMPUTER STORES
Master Charge and Bank Americard accepted.

Our Software is copyrighted and may not be reproduced or sold.

Add $1.50 per volume handling, all domestic shipments sent U.P.S. except APO and P.O. Box which go parcel post. Foreign orders add $8.00/volume for air shipment and make payable in U.S. dollars only.

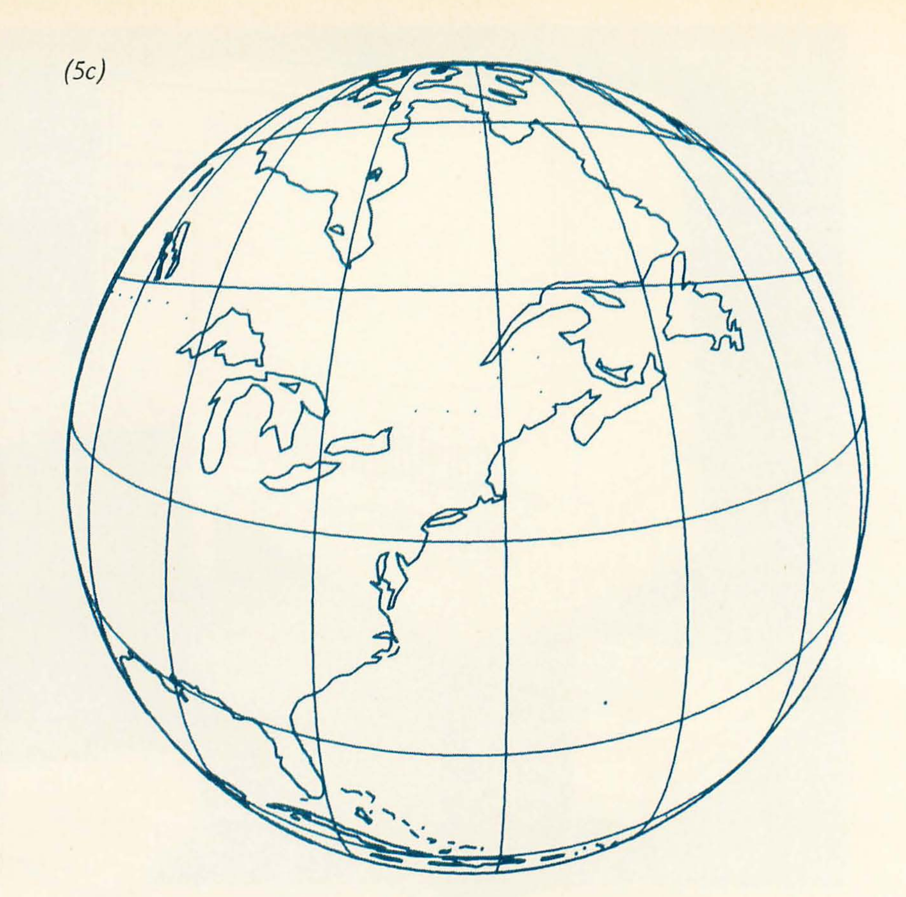


P.O. Box 490099-B Key Biscayne, FL 33149

Phone orders call 800-327-6543 Information — (305) 361-1153

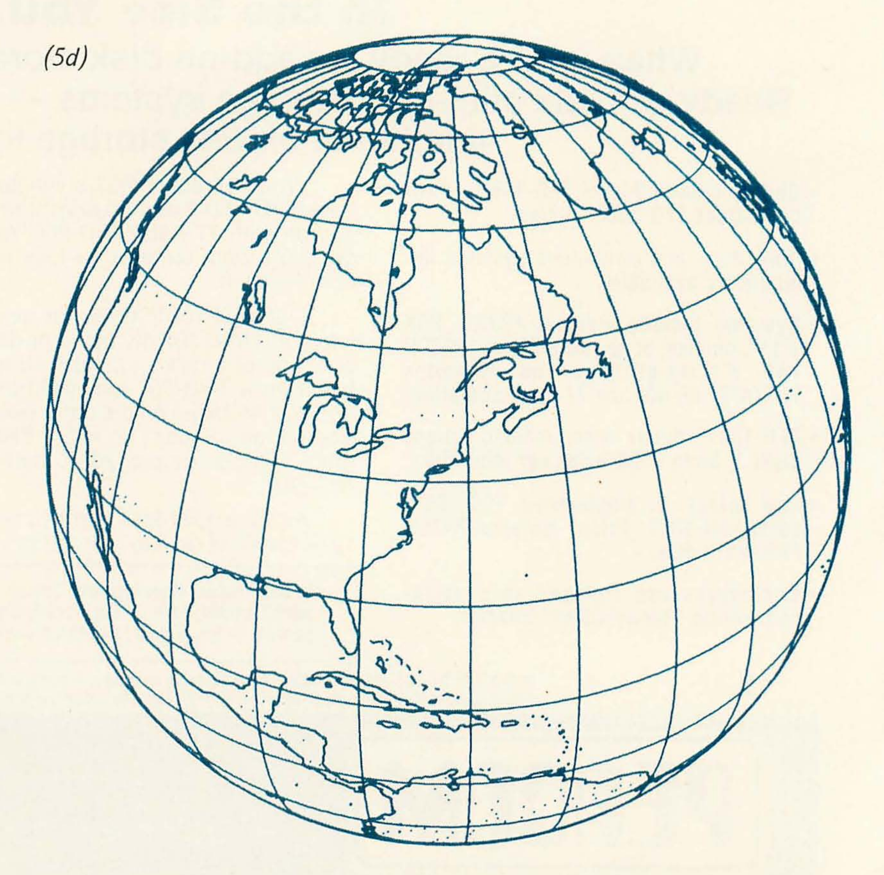
Circle 301 on inquiry card.

(5c)



VIEWED FROM 1000 KM OVER 42:53N 71:57W, FACING 0 DEG AZIMUTH
PERSPECTIVE PROJECTION PREPARED BY WILLIAM D. JOHNSTON

(5d)

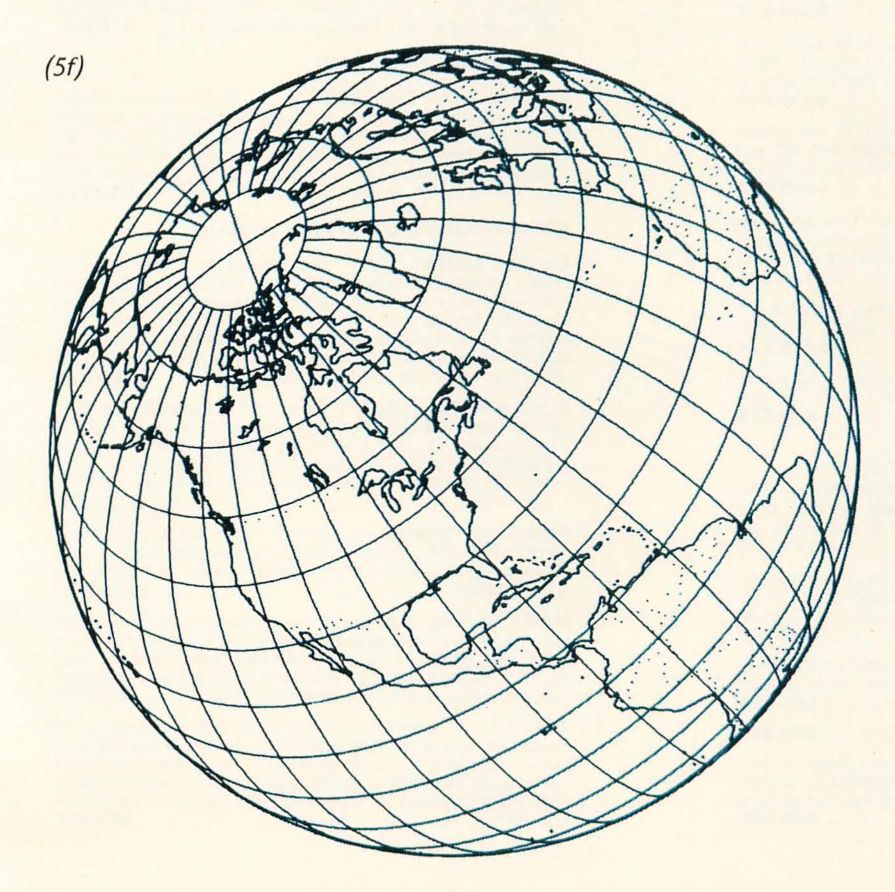


VIEWED FROM 5000 KM OVER 42:53N 71:57W, FACING 0 DEG AZIMUTH
PERSPECTIVE PROJECTION PREPARED BY WILLIAM D. JOHNSTON

(5e)

VIEWED FROM INFINITY. OVER 42:53N 71:57W. FACING 0 DEG AZIMUTH
PERSPECTIVE PROJECTION PREPARED BY WILLIAM D. JOHNSTON

(5f)



VIEWED FROM INFINITY. OVER 42:53N 71:57W. FACING 45 DEG AZIMUTH
PERSPECTIVE PROJECTION PREPARED BY WILLIAM D. JOHNSTON

Software for most popular 8080/Z80 computer disk systems, including ***NORTH STAR, MICROPOLIS, iCOM, SD SYSTEMS, DYNABYTE DB8/2, HELIOS, ALTAIR, TRS-80, 8" IBM*** *and* ***OHIO SCIENTIFIC*** *formats.*

Software with Manual / Manual Alone

GRAHAM-DORIAN SOFTWARE SYSTEMS

☐ **PAYROLL SYSTEM** — Maintains employee master file. Computes payroll withholding for FICA, Federal and State taxes. Prints payroll register, checks, quarterly reports and W-2 forms. Can generate ad hoc reports and employee form letters with mail labels. Requires CBASIC. Supplied in source code. .... **$605/$35**

☐ **APARTMENT MANAGEMENT SYSTEM** — Financial management system for receipts, disbursements and security deposits of apartment projects. Captures data on vacancies, revenues, etc. for annual trend analysis. Daily report shows late rents, vacancy notices, vacancies, income lost through vacancies, etc. Requires CBASIC. Supplied in source code. .... **$605/$35**

☐ **INVENTORY SYSTEM** — Captures stock levels, costs, sources, sales, ages, turnover, markup, etc. Transaction information may be entered for reporting by salesman, type of sale, date of sale, etc. Reports available both for accounting and decision making. Requires CBASIC. Supplied in source code. .... **$605/$35**

OTHER

☐ **Z80 Development Package** — Consists of: (1) disk file line editor, with global inter and intra-line facilities; (2) Z80 relocating assembler, Zilog/Mostek mnemonics, conditional assembly and cross reference table capabilities; (3) linking loader producing absolute Intel hex disk file for CP/M LOAD, DDT or SID facilities. Standard CP/M and TRS-80 CP/M versions available .... **$95/$15**

☐ **TEXTWRITER II** — Text formatter to justify and paginate letters and other documents. Special features include insertion of text during execution from other disk files or console, permitting recipe documents to be created from linked fragments on other files. Ideal for contracts, manuals, etc. .... **$75/$5**

☐ **DISINTEL** — Disk based disassembler to Intel 8080 or TDL/Xitan Z80 source code, listing and cross reference files. Intel or TDL/Xitan pseudo ops optional. Runs on 8080. Standard CP/M and TRS-80 CP/M versions available .... **$65/$10**

☐ **DISZILOG** — As DISINTEL to Zilog/Mostek mnemonic files. Runs on Z80 only. Standard CP/M and TRS-80 CP/M versions available .... **$65/$10**

Software with Manual / Manual Alone

☐ **WHATSIT?** — Interactive data-base system using associative tags to retrieve information by subject. Hashing and random access used for fast response. Requires CBASIC .... **$125/$25**

☐ **XYBASIC** Interactive Process Control BASIC — Full disk BASIC features plus unique commands to handle bytes, rotate and shift, and to test and set bits. Available in Integer, Extended and ROMable versions.
Integer Disk or Integer ROMable .... **$295/$25**
Extended Disk or Extended ROMable .... **$395/$25**

☐ **SMAL/80** Structured Macro Assembled Language — Package of powerful general purpose text macro processor and SMAL structured language compiler. SMAL is an assembler language with IF-THEN-ELSE, LOOP-REPEAT-WHILE, DO-END, BEGIN-END constructs .... **$75/$15**

☐ **Selector II** — Data Base Processor to create and maintain single Key data bases. Prints formatted, sorted reports with numerical summaries. Available for Microsoft and CBASIC (state which). Supplied in source code .... **$195/$20**

☐ **Selector III** — Multi (i.e., up to 24) Key version of Selector II. Comes with applications programs including Sales Activity, Inventory, Payables, Receivables, Check Register, Expenses, Appointments, and Client/Patient. Requires CBASIC Supplied in source code .... **$295/$20**
Enhanced version for CBASIC-2 .... **$345/$20**

☐ **CPM/374X Utility Package** — has full range of functions to create or re-name an IBM 3741 volume, display directory information and edit the data set contents. Provides full file transfer facilities between 3741 volume data sets and CP/M files .... **$195/$10**

☐ **Flippy Disk Kit** — Template and instructions to modify single sided 5¼" diskettes for use of second side in singled sided drives **$9.75**

☐ **BASIC Comparison** — A comprehensive features and performance analysis of five 8080 disk BASIC languages — CBASIC, BASIC-E, XYBASIC, Microsoft Disk Extended BASIC, and Xitan's Disk BASIC. Itemizes results of 21 different benchmark tests for speed and accuracy and lists instructions and features of each BASIC ......(send 20¢ S.A.S.E.) **FREE**

☐ **TRS-80 FORTRAN PACKAGE** — Professional disk-based language and utility package written by Microsoft, creators of Level II BASIC, the package runs on a TRS-80 system with 32K RAM, one or more drives and TRSDOS. The software is supplied on diskettes and consists of a relocatable machine code FORTRAN Compiler, Macro Assembler, a Linking Loader, Subroutine Library, Text Editor **$325**
Macro assembler, loader and editor alone **$165**

Fill it in.

Lifeboat Associates, 2248 Broadway, New York, N.Y. 10024 **(212) 580-0082**

Software	Price
☐ manual alone	
☐ manual alone	
☐ Check ☐ U.P.S. COD ☐ Visa ☐ Master Charge — Shipping	
Account # Exp. Date — $1.00 for C.O.D.	
Signature — Total	

My computer configuration (specifying disk system):

Name

Address (No P.O. Box)

City State Zip

EFFECTIVE APRIL 15, 1979

TM The Software Supermarket is a trademark of Lifeboat Associates

All software requires CP/M for operation except TRS-80 FORTRAN

Disk systems and formats: North Star single or double density, IBM single or 2D/256, Altair, Helios II, Micropolis Mod I or II, 5¼" soft sector (Micro iCOM/SD Sales/ Dynabyte), etc.

Add $1/item shipping ($2 min.). Add $1 additional for UPS C.O.D.

Manual cost applicable against price of subsequent software purchase.

The sale of each proprietary software package conveys a license for use on one system only.

Lifeboat Associates
THE SOFTWARE SUPER-MARKET
TM

41:58N 106:47W
AZIMUTHAL EQUIDISTANT PROJECTION PREPARED BY WILLIAM D. JOHNSTON

Figure 6: This azimuthal equidistant map is basically the same as those illustrated in part 1. We have added a grid of meridians and parallels to emphasize geographic distribution.

Text continued from page 114:

was facing north to begin with). The program permits the view to be rotated by any amount and for any map, regardless of altitude. As mentioned earlier, this feature is particularly useful when the map is being generated on a video terminal.

Azimuthal Equidistant Maps

Figure 6 illustrates a map that is very similar to the azimuthal equidistant (great circle) maps shown in part 1. The primary difference is that we have added a grid of meridians and parallels to figure 6 to give more meaning to geographic distribution. This map projection is extremely useful when applied to such fields as navigation and radio communication, but you must not forget the inherent distortions. The grid system on this map helps dramatize where these distortions lie.

Another azimuthal equidistant projection is presented in figure 7, but the coverage is limited to 90° of arc (half that of the map in figure 6). This not only expands the scale of the map, but it eliminates the portion with the greatest distortion. The example is of an Oscar satellite communications coverage map centered on a location near Geneva, Switzerland. By taking advantage of the fact that radial distances (and, hence, arc distances) from the center are linear in this type of projection, it is a simple matter to draw a circle that will indicate the maximum possible communication range through a given satellite. The central location (Geneva, in this case) can then communicate to any location within the circle, at some time or another, depending upon the satellite's position.

The map, in this example, has been overlaid with three different circles to show the maximum range for all of the currently operational satellites (Oscar 7, Oscar 8, and RS:1–2, starting with the inner circle and moving outward). Note the difference

between this map and those in figures 3 and 4. The map in figure 7 shows *all* locations with which communication is possible, irrespective of satellite position, whereas the maps of figures 3 and 4 show where communication is possible at some particular moment in time when the satellite is over a given point.

Summary

In part 1 we discussed the fundamental methods and resources required to produce any kind of map on a computer. We also presented several simple programs in BASIC, each containing only about a dozen executable statements, but which are capable of producing a number of attractive and useful map projections.

In this conclusion to "Computer Generated Maps", we have shown both the need and the means to develop a single, general purpose, map projection program with the flexibility to produce a variety of perspective and azimuthal equidistant maps. The subroutine given in listing 1 is an efficient, functional program which does just that, yet it only contains about 60 executable statements. All of the maps illustrated in this article were produced by that program, and they are only a sampling of its total capability. Whether you plan to generate maps for use with communications satellites, maps for the captain's console of a spacecraft, or maps for the sake of having maps, the program presented here can enhance your system's capability enormously and give you many hours of enjoyment.■

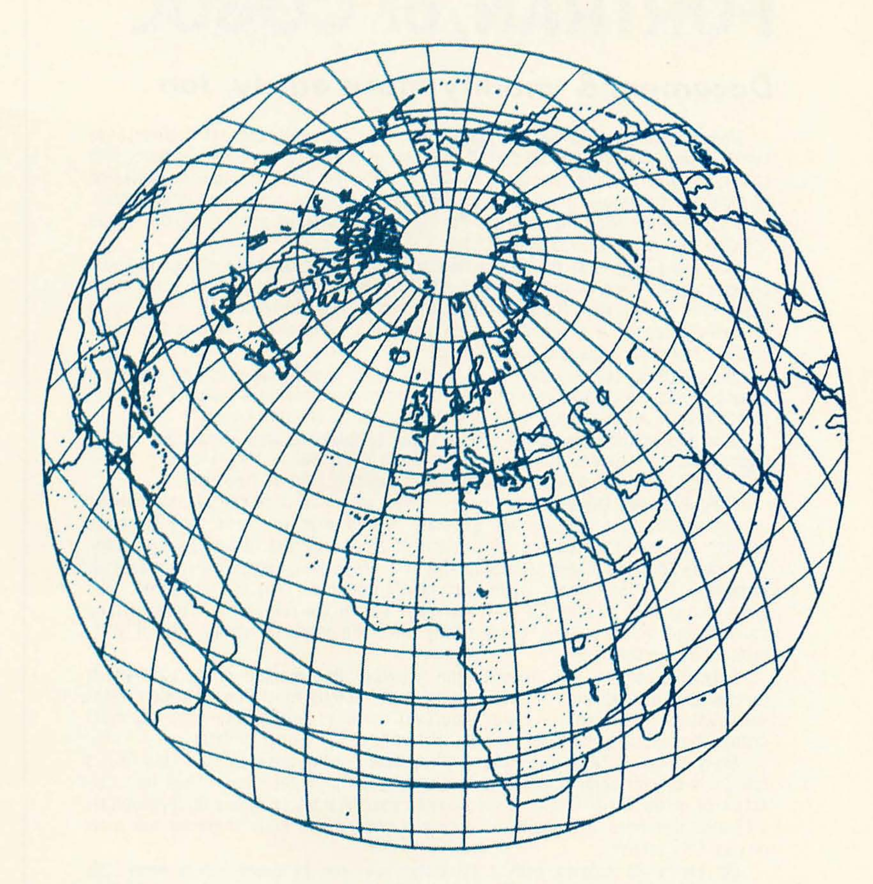


Figure 7: Another azimuthal equidistant projection is presented here, but we have limited the coverage to 90° of arc. This expands the scale of the map. The three concentric circles that have been superimposed on this map show the maximum possible communications range from the central location, through all of the currently operational amateur radio communications satellites (Oscar 7, Oscar 8, and RS:1–2).

Text continued from page 6:

will be a more elaborate set of read only memory code which interacts with the music keyboard and a special function keyboard to be defined. Limited timbre setups of the synthesizer can then be made when it is played and isolated from the data base facilities of the Pascal machine.

As for main software, capacity will be needed to execute some form of interpretive real time control software, possibly through the use of a threaded interpretive language somewhat like Forth. Implementations of this sort of language abound: Forth, Urth, IPS, and numerous unnamed homebrew versions. Such interpreters, which are fairly simple to code in machine language without an assembler, provide an excellent path to more significant software in a homebrew situation with a new processor. Everyone I know who uses them becomes a fanatic, so there are obviously some strong emotional arguments for flexibility and power that get people addicted. Why not try one?

Hardware reflecting this requirement will perhaps be 4 K bytes of read only memory for the kernel of the interpreter design. Software design and development will, of course, be done in machine language using the Pascal machine as a filing and program development tool.

In addition to the 6809 processor, the hardware of the new machine will probably include 32 K bytes of programmable memory in the low end of address space, 16 K bytes of 2708 read only memory sockets for the various segments of the detail low level software, a serial port for the communications interface, a parallel port for the synthesizer interface, a parallel port for the music keyboard and miscellaneous keyswitch inputs, several uncommitted parallel ports, and a parallel port for the Sykes floppy, borrowed from the older system to be used as a mass storage subsystem.

This new processor will reflect a number of the improvements that have been made in the experimenter's computer system art over the past few years. It will have a much smaller parts count due to the 16 K dynamic memory parts I intend to use and I will pay attention to packaging, as I want to be able to carry the results around. It will be entirely fabricated with convenient Vector Slit-N-Wrap interconnection, although I now use the motorized tool to minimize the chance of open connections which occurred when I used that method by hand. And, of course, there is the thrill of experimenting with a new processor, the 6809 design described by Terry Ritter and Joel Boney in recent issues of BYTE.■


BYTE
BOOKS
TM

...And the future

THE BYTE BOOK OF COMPUTER MUSIC combines the best computer music articles from past issues of BYTE Magazine with exciting new material—all written for the computer experimenter interested in this fascinating field.

You will enjoy Hal Chamberlin's "A Sampling of Techniques for Computer Performance of Music", which shows how you can create four-part melodies on your computer. For the budget minded, "A $19 Music Interface" contains practical tutorial information—and organ fans will enjoy reading "Electronic Organ Chips For Use in Computer Music Synthesis".

New material includes "Polyphony Made Easy" and "A Terrain Reader". The first describes a handy circuit that allows you to enter more than one note at a time into your computer from a musical keyboard. The "Terrain Reader" is a remarkable program that creates random music based on land terrain maps.

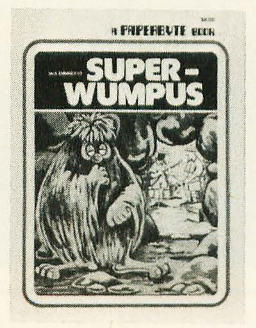
Other articles range from flights of fancy about the reproductive systems of pianos to Fast Fourier transform programs written in BASIC and 6800 machine language, multi-computer music systems, Walsh Functions, and much more.

For the first time, material difficult to obtain has been collected into one convenient, easy to read book. An ardent do-it-yourselfer or armchair musicologist will find this book to be a useful addition to the library.



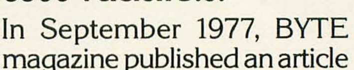
ISBN 0-931718-11-2
Editor: Christopher P. Morgan
Pages: approx. 128
Price: **$10.00**

SUPERWUMPUS is an exciting computer game incorporating the original structure of the WUMPUS game along with added features to make it even more fascinating. The original game was described in the book **What To Do After You Hit Return,** published by the People's Computer Company. Programmed in both 6800 assembly language and BASIC, SUPERWUMPUS is not only addictively fun, but also provides a splendid tutorial on setting up unusual data structures (the tunnel and cave system of SUPERWUMPUS forms a dodecahedron). This is a **PAPERBYTE**™ book.



ISBN 0-931718-03-1
Author: Jack Emmerichs
Pages: 56
Price: **$6.00**

TINY ASSEMBLER 6800, Version 3.1 is an enhancement of Jack Emmerichs' successful Tiny Assembler. The original version (3.0) was described first in the April and May 1977 issues of BYTE magazine, and later in the **PAPERBYTE**™ book **TINY ASSEMBLER 6800 Version 3.0.**



In September 1977, BYTE magazine published an article entitled, "Expanding The Tiny Assembler". This provided a detailed description of the enhancements incorporated into Version 3.1, such as the addition of a "begin" statement, a "virtual symbol table", and a larger subset of the Motorola 6800 assembly language.

All the above articles, plus an updated version of the user's guide, the source, object and PAPERBYTE™ bar code formats of both Version 3.0 and 3.1 make this book the most complete documentation possible for Jack Emmerichs' Tiny Assembler.

ISBN 0-931718-08-2
Author: Jack Emmerichs
Pages: 80
Price: **$9.00**

A walk through this book brings you into **Ciarcia's Circuit Cellar** for a detailed look at the marvelous projects which let you do useful things with your microcomputer. A collection of more than a year's worth of the popular series in BYTE magazine, **Ciarcia's Circuit Cellar** includes the six winners of BYTE's On-going Monitor Box (BOMB) award, voted by the readers themselves as the best articles of the month: **Control the World** (September 1977), **Memory Mapped IO** (November 1977), **Program Your Next EROM in BASIC** (March 1978), **Tune In and Turn On** (April 1978), **Talk To Me** (June 1978), and **Let Your Fingers Do the Talking** (August 1978).

Each article is a complete tutorial giving all the details needed to construct each project. Using amusing anecdotes to introduce the articles and an easy-going style, Steve presents each project so that even a neophyte need not be afraid to try it.



ISBN 0-931718-07-4
Author: Steve Ciarcia
Pages: approx. 128
Price: **$8.00**

is right now!

BASEX, a new compact, compiled language for microcomputers, has many of the best features of BASIC and the 8080 assembly language—and it can be run on any of the 8080 style microprocessors: 8080, Z-80, or 8085. This is a **PAPERBYTE™** book.

Subroutines in the **BASEX** operating system typically execute programs up to five times faster than equivalent programs in a BASIC interpreter—while requiring about half the memory space. In addition, **BASEX** has most of the powerful features of good BASIC interpreters including array variables. text strings, arithmetic operations on signed 16 bit integers, and versatile IO communication functions. And since the two languages, BASEX and BASIC, are so similar, it is possible to easily translate programs using integer arithmetic data from BASIC into BASEX.

The author, Paul Warme, has also included a BASEX Loader program which is capable of relocating programs anywhere in memory.



ISBN 0-931718-05-8
Author: Paul Warme
Pages: 88
Price: **$8.00**

PROGRAMMING TECHNIQUES is a series of **BYTE BOOKS** concerned with the art and science of computer programming. It is a collection of the best articles from BYTE magazine and new material collected just for this series. Each volume of the series provides the personal computer user with background information to write and maintain programs effectively.



The first volume in the Programming Techniques series is entitled **PROGRAM DESIGN.** It discusses in detail the theory of program design. The purpose of the book is to provide the personal computer user with the techniques needed to design efficient, effective, maintainable programs. Included is information concerning structured program design, modular programming techniques, program logic design, and examples of some of the more common traps the casual as well as the experienced programmer may fall into. In addition, details on various aspects of the actual program functions, such as hashed tables and binary tree processing, are included.

ISBN 0-931718-12-0
Editor: Blaise W. Liffick
Pages: 96
Price: **$6.00**

SIMULATION is the second volume in the Programming Techniques series. The chapters deal with various aspects of specific types of simulation. Both theoretical and practical applications are included. Particularly stressed is simulation of motion, including wave motion and flying objects. The realm of artificial intelligence is explored, along with simulating robot motion with the microcomputer. Finally, tips on how to simulate electronic circuits on the computer are detailed.

ISBN 0-931718-13-9
Editor: Blaise W. Liffick
Pages: approx. 80
Price: **$6.00**
Publication: Winter 1979

RA6800ML: AN M6800 RELOCATABLE MACRO ASSEMBLER is a two pass assembler for the Motorola 6800 microprocessor. It is designed to run on a minimum system of 16 K bytes of memory, a system console (such as a Teletype terminal), a system monitor (such as Motorola MIKBUG read only memory program or the ICOM Floppy Disk Operating System), and some form of mass file storage (dual cassette recorders or a floppy disk).

The Assembler can produce a program listing, a sorted Symbol Table listing and relocatable object code. The object code is loaded and linked with other assembled modules using the **Linking Loader LINK68.** (Refer to **PAPERBYTE™** publication **LINK68: AN M6800 LINKING LOADER** for details.)

There is a complete description of the 6800 Assembly language and its components, including outlines of the instruction and address formats, pseudo instructions and macro facilities. Each major routine of the Assembler is described in detail, complete with flow charts and a cross reference showing all calling and called-by routines, pointers, flags, and temporary variables.

In addition, details on interfacing and using the Assembler, error messages generated by the Assembler, the Assembler and sample IO driver source code listings, and **PAPERBYTE™** bar code representation of the Assembler's relocatable object file are all included.

This book provides the necessary background for coding programs in the 6800 assembly language, and for understanding the innermost operations of the Assembler.

ISBN 0-931718-10-4
Author: Jack E. Hemenway
Pages: 184
Price: **$25.00**

to order books see next page .....

LINK68: AN M6800 LINKING LOADER is a one pass linking loader which allows separately translated relocatable object modules to be loaded and linked together to form a single executable load module, and to relocate modules in memory. It produces a load map and a load module in Motorola MIKBUG loader format. The Linking Loader requires 2 K bytes of memory, a system console (such as a Teletype terminal), a system monitor (for instance, Motorola MIKBUG read only memory program or the ICOM Floppy Disk Operating System), and some form of mass file storage (dual cassette recorders or a floppy disk).

It was the express purpose of the authors of this book to provide everything necessary for the user to easily learn about the system. In addition to the source code and **PAPERBYTE™** bar code listings, there is a detailed description of the major routines of the Linking Loader, including flow charts. While implementing the system, the user has an opportunity to learn about the nature of linking loader design as well as simply acquiring a useful software tool.

ISBN 0-931718-09-0
Authors: Robert D. Grappel
& Jack E. Hemenway
Pages: 72
Price: **$8.00**
Winter 1979

TRACER: A 6800 DEBUGGING PROGRAM is for the programmer looking for good debugging software. TRACER features single step execution using dynamic break points, register examination and modification, and memory examination and modification. This book includes a reprint of "Jack and the Machine Debug" (from the December 1977 issue of BYTE magazine), **TRACER** program notes, complete assembly and source listing in 6800 assembly language, object program listing, and machine readable **PAPERBYTE™** bar codes of the object code.

ISBN 0-931718-02-3
Authors: Robert D. Grappel
& Jack E. Hemenway
Pages: 24
Price: **$6.00**

MONDEB: AN ADVANCED M6800 MONITOR-DEBUGGER has all the general features of Motorola's MIKBUG monitor as well as numerous other capabilities. Ease of use was a prime design consideration. The other goal was to achieve minimum memory requirements while retaining maximum versatility. The result is an extremely versatile program. The size of the entire MONDEB is less than 3 K.

Some of the command capabilities of MONDEB include displaying and setting the contents of registers, setting interrupts for debugging, testing a programmable memory range for bad memory locations, changing the display and input base of numbers, displaying the contents of memory, searching for a specified string, copying a range of bytes from one location in memory to another, and defining the location to which control will transfer upon receipt of an interrupt. This is a **PAPERBYTE™** book.

ISBN 0-931718-06-6
Author: Don Peters
Pages: 88
Price: **$5.00**

BAR CODE LOADER. The purpose of this pamphlet is to present the decoding algorithm which was designed by Ken Budnick of Micro-Scan Associates at the request of BYTE Publications, Inc., for the **PAPERBYTE™** bar code representation of executable code. The text of this pamphlet was written by Ken, and contains the general algorithm description in flow chart form plus detailed assemblies of program code for 6800, 6502 and 8080 processors. Individuals with computers based on these processors can use the software directly. Individuals with other processors can use the provided functional specifications and detail examples to create equivalent programs.

ISBN 0-931718-01-5
Author: Ken Budnick
Pages: 32
Price: **$2.00**

BYTE BOOKS Division • 70 Main Street • Peterborough, New Hampshire 03458

Name ____ Title ____ Company ____

Street ____ City ____ State/Province ____ Code ____

☐ Check enclosed in the amount of $ ____
☐ Bill Visa ☐ Bill Master Charge Card No. ____ Exp. Date ____

Please send the books I have checked.

- ☐ **Computer Music $10.00**
- ☐ **SUPERWUMPUS $6.00**
- ☐ **Tiny Assembler (3.1) $9.00**
- ☐ **Circuit Cellar $8.00**
- ☐ **BASEX $8.00**
- ☐ **Program Design $6.00**
- ☐ **Simulation $6.00**
- ☐ **RA6800ML $25.00**
- ☐ **Link68 $8.00**
- ☐ **TRACER $6.00**
- ☐ **Mondeb $5.00**
- ☐ **Bar Code Loader $2.00**

Add **60¢** per book to cover postage and handling

Please allow 6–8 weeks for processing your order.

Circle 36 on inquiry card.

BYTE News . . . .

NCC/NYC TO BE THE BIGGEST COMPUTER SHOW EVER. The National Computer Conference (NCC) will happen again June 6 thru 9. Last year 57, 224 attendees turned out for the show, held in Anaheim CA. This year the NCC will be held in the New York City Coliseum. AFIPS, the sponsoring organization, expects attendance to top that of last year. Approximately 400 companies have reserved 1,700 booth spaces on four floors of the Coliseum, with overflow at the New York Hilton and Americana hotels. Last year 396 companies occupied 1,400 booths.

NCC will have a personal computing adjunct at the Americana Hotel, a few blocks away. It will probably be played down, as it was last year. By way of example, the personal computing exhibitors and speakers were not listed in the regular show program book handed out to each attendee; hence, many attendees last year were unaware of the personal computing part of the show.

S-100 BUS STANDARD TO BE ADOPTED SOON. An IEEE committee has been working on a standard for the S-100 bus for over a year, and adoption is expected very soon. Much of the credit for this standard goes to George Morrow of Thinker Toys.

This standard will do two things. One, it will resolve the conflicts between the use of many bus pins by different manufacturers and eliminate the lack of compatibility between many "S-100 compatible" plug-in boards. Two, and possibly more important, it provides use of the S-100 bus for 16 bit processors for extended addressing of up to 8 M bytes of memory and for master-slave multiprocessor systems. This will make the S-100 bus the most powerful bus around and will, no doubt, continue and increase its popularity.

TI AND HP PC SYSTEMS RUMORS. Texas Instruments and Hewlett-Packard continue to maintain tight lips on their rumored personal computer systems. As TI has said, "TI will not discuss products that have not yet been announced." However, information has leaked out on these units which are expected to have a tremendous impact on the personal computing market. Several rumors have been reported in previous BYTE NEWS columns. The latest is that TI will introduce their entry at either the NCC show in June or the Consumer Electronics Show in July. In either event, it is expected to be ready for the 1979 Christmas market.

The HP computer is also expected to be ready by Christmas, and is anticipated to be a stripped down version of their current table-top system. This means that it will use BASIC and be expandable.

Both HP and TI are expected to have $500 list prices for the basic unit. Key retailers have already been approached by both TI and HP to set up a selective distribution. It is rumored that they will favor selected personal computing stores that can do justice to software requirements.

INTEL TO PRODUCE ANALOG MICROPROCESSOR AND SUPER 8 BIT MICROPROCESSORS. Real time processing of analog signals by microprocessors has been severely limited by the slow speed of most microprocessors. For example, an 8080 clocked at 2 M Hz can, at best, synthesize clean sine waves at about 1 to 2 k Hz, which is the low end of the audio spectrum. This fall, Intel will introduce an integrated circuit which combines an analog-to-digital converter, a digital-to-analog converter, microprocessor and read only memory on a single device. It will be capable of processing analog signals up to 13 k Hz. Called the 2920, the integrated circuit will have a 9 bit conversion register. It could be used in conjunction with an 8080 processor, where the 2920 does the signal processing while the 8080 does the data processing.

Intel has done another clever thing. They have taken an 8086 and limited its data I/O (input/output) to 8 bits and memory addressing to 16 bits. It is called the 8088 and will deliver five times the performance of the 8080 (2 M Hz). Actually, the 8088 is an 8086 split into two 8 bit microprocessors on one integrated circuit, one handling I/O and the other data processing. It offers most of the features of the 8086 (eg: hardware multiply/divide).

NATIONAL SEMICONDUCTOR TO INTRODUCE NEW MICROS. It is nearly three years since National introduced their last microprocessor. (Actually, we must give National credit for pioneering the 16 bit microprocessor with the PACE and IMP-16 microprocessors introduced in 1975.) Now National is bringing out a new CMOS 8 bit microprocessor that will be software compatible with the 8080, have added features and consume less power. Further, they will introduce a 16 bit microprocessor that is a "cut above" the Z-8000 and 68000. Production is expected by the end of the year.

MICROPROCESSORS FOR $1 APIECE? Maybe not this year. . .but it is approaching fast. Synertek recently reduced the 100 lot price for the 6502 (used in the PET, Apple, OSI, etc) from $10 to $7. In high volume they have reduced the price from $4 to $2.50. I can still remember paying $350 for an 8080, just four years ago!

DOUBLE SIDED DISK DRIVES STILL IN LIMITED PRODUCTION. Despite advertising and promotion for double sided disk drives, most makers are still having trouble gearing up from prototype runs to full production. Shugart Associates, for example, does not expect to be in normal production until the end of the year.

COLOR VIDEO TERMINALS COMING ON STRONG. The prices of color video terminals, which until now have been very expensive as compared to black and white video terminals, are expected to decrease substantially within the coming year. Further, they will have more features. The reasons for the price decrease are cost reduction in electronics and increased production, as demand increases. More businesses are finding that the difference in price for color is worth it in many applications, an example being Management Information Systems. We can expect low cost color video terminals on the market for under $1,000.

At the same time, manufacturers are developing driver software for video terminals which exploit the color capability, in particular, combining graphics and alphanumerics. One example would be bit map routines allowing the creation of multiple graphics regions on the video terminal while having alphanumeric regions.

PASCAL NOW AVAILABLE FOR 6800. All 6800 owners who have been envious of the Pascal that is available to other processor users can now have their own Pascal. Control Systems Inc, Kansas City KS, has just introduced a 6800 version of the UCSD Pascal, Version II.

HOW ABOUT A COMPUTER VACATION? Want to combine vacation and hobby? A group of 20 to 50 personal computerists are doing just that during Christmas week. They have organized a weeklong workshop to be held at a Caribbean resort. Families are welcome. If you are interested in participating, write either Dr Andy Bender, 400 Old Hook Rd, Westwood NJ 07675, or Dr Jeff Brownstein, 2 Tor Rd, Wappinger Falls NY 12590.

QUIP VERSUS THE DIP. The new 16 bit microprocessors and the 32 bit microprocessors on the drawing boards have created packaging problems for integrated circuit makers. How are they going to get all addressing, data, I/O (input/output) lines on an integrated circuit package? Anyone who handles 40 pin dual-in-line packages knows the handling problem. Well, Intel and 3M have jointly developed a new 64 pin integrated circuit header called QUIP (quad-in-line package). It will have two rows of 16 pins along both edges of the package and will shrink the package from 3 1/8 inches, for a 64 pin dual-in-line package (DIP) to 1 5/8 inches for a QUIP. Further, internal lead paths will decrease, reducing capacitance, resistance and inductance, and allowing higher operating speeds. Pins will still be on 0.1 inch centers and the QUIP will cost 15 percent less than the DIP.

4 K BYTE PROGRAMMABLE MEMORY INTRODUCED. As the size of memory circuits increases, integrated circuit makers are going the byte-size memory route to afford easier interfacing to microprocessor buses. Zilog is the first to introduce an 8 bit byte programmable memory. Called the Z6132, it is a 32 K bit memory organized as 4 K words by 8 bits. It uses a one transistor memory cell and includes on-chip refresh control circuitry. Pinout permits easy use in 16 bit systems. Hardware keeps getting easier!

DOD LOVES ADA. After years of trying to standardize a high level computer language, the United States Department of Defense (DOD) has created a special group for this purpose. The final approved language will be called ADA, after Ada Augusta, Countess of Lovelace, who is credited with being the very first programmer.

The DOD have narrowed their choice to two different Pascal-like designs. The preliminary design is due for delivery in May of this year, with final approval expected at the end of the year.

LIQUID CRYSTAL DISPLAYS IMPROVING. The graphics terminal marketplace is going to see some radical changes within the next few years as new technologies develop to compete with the age old video type graphics terminal technology. Plasma panel and liquid crystal display elements (LCDE) are examples of a new graphics technology. The plasma panels offer higher brightness, no flicker, and touch sensing ability. The LCDE are created by a scanning laser beam. The LCDE have their own memory, and do not have to be refreshed. Further, they offer very high resolution, color and projection capabilities. Both Western Electric and IBM are doing developmental work in this area.

DEC OPENS SECOND RETAIL STORE. Digital Equipment Corp recently opened its second store in Boston. Their first store was opened in New Hampshire in July of last year. More stores are planned for "off the shelf" purchases. Further, the stores will offer services such as mail list generation and word processing.

COMPUTER STORES' FUTURE APPEARS EXCELLENT. According to a recently completed study by marketing research firm Frost and Sullivan, the main distribution channel for microcomputer systems will continue to be the dedicated computer store, for many years. The hobby business will continue to increase, but at a lower rate. The larger increase will come from small business purchases.

Sol Libes
ACGNJ
1776 Raritan Rd
Scotch Plains NJ 07076

The Nature of Robots

Part 1: Defining Behavior

William T Powers
1133 Whitfield Rd
Northbrook IL 60062

A scientific revolution is just around the corner, and anyone with a personal computer can participate in it. The last time this happened, 250 years ago, the equipment was the homebrew telescope and the subject was astronomy. Now, astronomy belongs just as much to amateurs as to professionals. This time the particular subject matter is human nature and in a broader scope, the nature of all living systems. Some ancient and thoroughly accepted principles are going to be overturned, and the whole direction of scientific investigation of life processes will change.

The key concept behind this revolution is control theory. Control theory has been developing for almost 40 years, and has already been proposed (by Norbert Wiener) as a revolutionary concept. It has not been easy, however, to see just how control theory can be made part of existing scientific approaches although many people have tried. Most of these attempts have tried to wedge control theory into existing patterns of thought. To apply any new idea in such a way, while ignoring the new conceptual scheme made possible, is to deny the full potential of the new idea.

Many life scientists who have tried to use control theory have tried to imitate the engineering approach, dealing with human beings as part of a man-machine system instead of complete control systems in their own right. Others have used control theory directly to make models of human and animal behavior, but have concentrated on minor subsystems, failing to see that the organism as a whole can be dealt with in terms of the same principles. The result has often been a strange mixture of concepts — a patchwork instead of a system.

Strangely enough, many engineers who do understand control theory haven't done much better. Here the problem is that these engineers tend to accept the basic concepts developed by biologists and psychologists, and to use control theory to explain cause-effect relationships they are told exist — but which in fact do not exist. We will start this development by looking at something called *behavior,* which biologists and psychologists have assured engineers is very important, thereby leading the engineers astray.

What is all this supposed to mean? A lot is meant, though in different ways. Roboticists, for example, are trying to develop machines which will imitate human organization, and so are the artificial intelligence experimenters. But from whence came the description of the system they are trying to model? Basically, it came from the life sciences. If the life sciences are using the wrong model, it would be essential to know that before much more labor is invested in imitating an imaginary creature.

Perhaps the most general reason control theory is interesting is that it concerns people. There aren't many sciences left in which important discoveries can be made by amateurs working at their own tables. Control theory opens up an entirely new field of experimentation, a kind that has never been done before in psychology or any other life science.

All that is needed by amateurs who want to participate in these developments is a basic grasp of control theory, an understanding of the procedures that go with it, some basic equipment, and curiosity about human nature. I shall now provide the first two items on that list. The rest is up to you.

The Problem With Behavior

The word *behavior* is used frequently — we hear about behavioral science, behavior modification, behavior therapy. For example, *Science News* now has a "Behavior Column";

BYTE's Bugs

A Negative Sine

The arcsine and arccosine routines discussed in "Inverse Trig Functions" by Alan Miller (March 1979 BYTE, page 92) will not work for negative values of X. For arcsine, I recommend (in Mr Miller's notation):

```
DEF FNSN(X) = ATN(X/SQR(1-X*X))
```

and for cosine:

```
DEF FNCS(X) =
1.570796327-ATN(X/SQR(1-X*X)).
```

The constant 1.570796327 is, of course, $\pi/2$. These routines give the correct principal value for any value of X with an absolute value of less than 1.

John A Ball
Oak Hill Rd
Harvard MA 01451■

BYTE's Bits

Robot Information

James A Gupton Jr, author of "Talk to a Turtle" which appears in this issue of BYTE, has offered the following additional sources for robot information:

International Institute for Robotics
POB 615
Pelahatchie MI 39145
Attention: Dale Cowsert, Director

Offers a complete robot correspondence course including a microcomputer and a 2½ foot working robot. Costs range from Mentor II at $890 to Omnidex I at $3100.

Gallaher Research Inc
POB 10767 Salem Station
Winston-Salem NC 27108

Offers catalog on robot components to build any size robot from one to ten feet tall. Catalog cost is $10.■

it was formerly the publication's "Psychology Column". An innocent bystander might conclude that any word this important must have a universally accepted definition, but that is not true. Behavior is a slippery concept.

Here is an example of a person behaving. Chip Chad is seated in front of a teletypewriter pounding keys. What is he doing?

Is he alternately tensing and relaxing muscles in his arms? Yes. Is he moving his fingers up and down? Yes. Is he typing strings of symbols? Yes. Is he adding a return instruction that he forgot at the end of a subroutine? Yes. Is he writing a program for plotting stock market prices? Yes. Is he making a little extra money for a vacation? Yes. Is he justifying his hobby to his family? Yes.

Clearly, each description of what Chip is doing is, in fact, an accurate description of the very same collection of actions. Which one, then, is Chip's behavior? Obviously, they all are expressions of behavior.

Suppose Chip decides that he really doesn't need a subroutine, and substitutes a jump instruction for the return. Now, he is writing the program – obviously the same program – by using a different behavior. Or suppose he buys an input device, and continues working on the subroutine by speaking letters into a microphone. Now he is using different muscles and movements, but he is still doing the same behaviors farther down the list. How could he be doing the same thing by means of doing something different?

Or consider Chip driving a car along a straight road. He is consciously steering. This happens to be a gusty March day, and every five minutes the wind changes speed and direction. Chip is an experienced driver, and continues to steer the car down the road in a straight line. If we look at what his arms are doing, however, we find that they are moving the steering wheel in an apparently random pattern, now centered, now far to the right, now far to the left. Somehow he is managing to produce a constant steering-the-car behavior by means of a behavior that is widely varying. The path of the car doesn't correlate with the position of the steering wheel at all.

Scientists have always thought of behavior as the final product of activity inside the organism. The brain sends commands to the muscles, which create forces, which produce movements, which generate the stable and repeatable patterns we recognize as behavior. There is, in principle, a chain of cause and effect, with the events at the end of the chain being caused by the events at the beginning. Such scientists would say that in the example with Chip at the computer keyboard, we were simply attending to various stages in that chain.

How does that picture fit in with Chip's driving the car in a straight line? The direction in which the car is going is affected by his movements of the steering wheel, and is farther out along the chain of causes and effects. But the wind adds its effects on the direction of the car *after* Chip's effects in the chain. Somehow he is varying his actions so that when their effects are added to the effects of the randomly varied wind, the result is something constant. If we had been thinking of driving the car in a straight line as Chip's behavior, we have to revise that idea: the direction of the car depends just as much on the wind as on Chip.

It may seem that we have simply moved our definition of behavior closer to Chip. But consider *how* he moves the steering wheel. The wheel moves when the forces reflected from the front wheels do not exactly balance the forces created by his muscles. As the car goes along, the roadbed tilts and various bumps and dips cause changes in the reflected forces. The wheel may be turned far to the right, into the crosswind, on the average, but maintaining the wheel in that position requires that his muscles be constantly changing tension, as the reflected steering wheel forces fluctuate. We have the same problem as before: Chip produces a varying output that affects the steering wheel, but the steering wheel is also being affected by forces that are independent of what Chip is doing with his muscles. Yet the *sum* of the muscle forces and those extraneous forces is zero, except when the steering wheel is changing position.

Even if we back up another step and call Chip's muscle tensions his behavior, we have trouble. Muscles are made to contract by signals from the nervous system, but muscles don't respond the same amount to a given signal every time they are used. They fatigue; other muscles interfere with them; joint angles change so that a given muscle tension can produce different amounts and directions of force. The only *behavior* that Chip produces which can be attributed entirely to Chip and not in part to his environment consists of the nerve signals that leave his nervous system and enter his muscles.

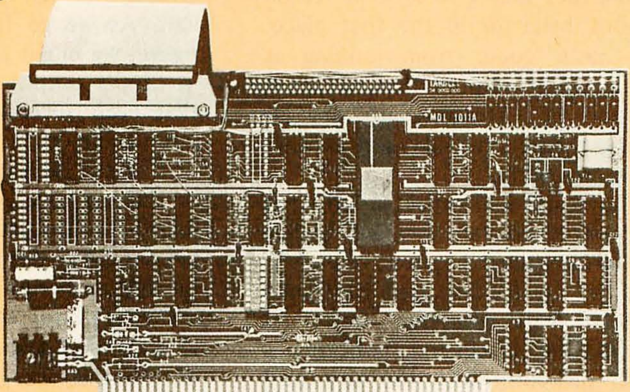
If we want to be completely accurate about Chip's behavior, we should consider the output signals from his nervous system, and leave everything else in his environment. That is what we will do, but by doing that we create the biggest problem of all.

A scientist studying a behavior hopes to learn enough about its rules to predict when

The TARBELL Connection

In an effort to offer products that meet the continually changing demands of the microcomputer industry, TARBELL ELECTRONICS is pleased to offer immediate delivery of these quality components and operating software. All TARBELL products are available from computer store dealers everywhere.

Tarbell Floppy Disk Interface



- Plugs directly into your IMSAI or ALTAIR and handles up to 4 standard single drives in daisy-chain.
- Operates at standard 250K per second on normal disk format capacity of 256K bytes.
- Works with modified CP/M Operating System and BASIC-E Compiler.
- Hardware includes 4 extra IC slots, built-in phantom bootstrap and on-board crystal clock. Uses WD 1771 LSI chip.
- Full 6-month warranty and extensive documentation.
- Kit $190 .... Assembled $265.

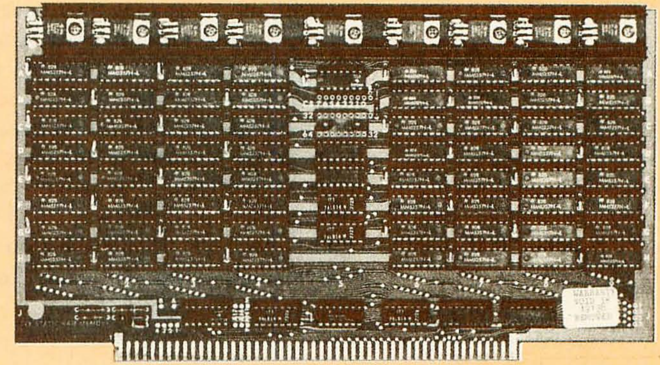
Specify drive for assembled units. Complete disk subsystems with operating software available. Please inquire for details.

Tarbell Disk BASIC

- Runs on 8080, 8085 or Z80
- Searches a file quickly for a string.
- Up to 64 files open at once.
- Random Access.
- Assignment of I/O.
- Alphanumeric line labels allowed.
- Read and Write string or numeric data.
- Unlimited length of variable names and strings.
- Procedures with independent variables.
- Number system 10 digits BCD integer or floating point.
- Chain to another program.
- Cause programs to be appended onto programs already in memory.
- Cause interpreter to enter edit mode using 15 single character edit commands.

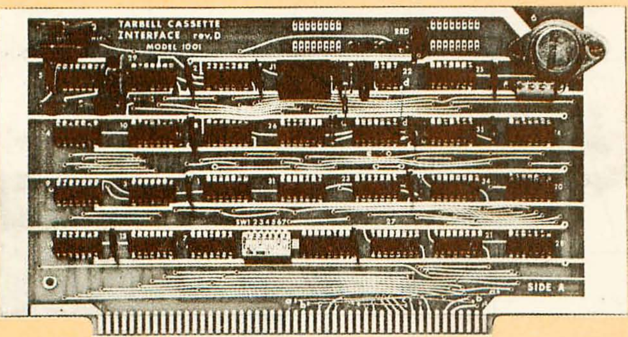
Occupies 24K of RAM. Tarbell BASIC on CP/M Disk .... $48.
Source on paper or CP/M Disk . . . $25.
CP/M and BASIC-E on disk with manuals .... $100.

Tarbell 32K RAM Memory



- 32K Static Memory
- S-100 Bus Connector
- 9 regulators provide excellent heat distribution.
- Extended addressing (bank switching.)
- Phantom line.
- Low power requirement.
- 20-Page operating manual.
- Full 1-year warranty.
- Assembled and tested full price only $625
- 16K version also available, assembled and tested only $390.

Tarbell Cassette Interface



- Plugs directly into your IMSAI or ALTAIR.
- Fastest transfer rate: 187 (standard) to 540 bytes/second.
- Extremely reliable—Phase encoded (self-clocking).
- 4 extra status lines, and 4 extra control lines.
- 37-page manual included.
- Device code selectable by DIP-switch.
- Capable of generating Kansas City tapes.
- No modification required on audio cassette recorder.
- Complete kit $120 .... Assembled $175.
- Manual may be purchased separately .... $8.
- Full 6-month warranty on kit and assembled units.

Tarbell Cassette BASIC

Includes most features of ALTAIR Extended BASIC, plus these added features:

- Assignment of I/O.
- Alphanumeric line labels.
- Unlimited length of variable names and strings.
- Number system 10 digits BCD integer or floating point.
- Procedures with independent variables.
- Read and Write string data.
- Multi-file capability.

Full price with complete documentation .... $48.

Prepaid, COD, or cash only. California residents please add 6% sales tax.

ALTAIR is a trademark/tradename of Pertec Computer Corporation
CP/M is a trademark/tradename of Digital Research



950 DOVLEN PLACE • SUITE B
CARSON, CALIFORNIA 90746
(213)538-4251 • (213) 538-2254

Circle 360 on inquiry card.

it will occur. Under the old approach, this means varying factors in the environment and looking for behaviors that correlate with those variations. But if we try to describe behavior in terms of the output signals from the nervous system, all correlations disappear. Oh, maybe we have a knee jerk or a sneeze left over, but we have lost all the regularities that give us some reason to talk about behavior in the first place. We would never guess, from looking at Chip's neural signal outputs, that the result of them would be a straight path of a car that is being forced one way and another by a variable crosswind.

When you pause and reflect upon what has been covered so far, you will realize that we are already deep into control theory, even though we haven't discussed it by name yet. We have dealt with the subject as such because the discussion concerns a fundamental difficulty with the very concept of behavior, especially the concept that behavior is the final product of an organism's inner activities. As we see how this difficulty gets resolved, we will be forced into control theory no matter how we approach the solution. One reason biologists or psychologists have not developed control theory is that they have clung stubbornly to the idea that behavior is part of a causal chain that starts in the nervous system (or in stimuli that cause activity in the nervous system) and propagates outward from there according to physical laws of cause and effect. That is why people design robots in the same way, and why those robots have yet to behave in a way that is convincingly alive. In order to solve this problem instead of just brushing it aside, we have to admit that the causal chain in which people have believed for so long simply does not exist, and never has existed.

Figure 1 sums up the problem we are dealing with. At every stage of events following the outputs from Chip's nervous system *disturbances* come into play, adding to the effects that can be traced to the neural signals. As we go farther to the right of the figure, we might expect that any regularities in Chip's output signals would be lost (ie: that each successive variable would show more and more random variations).

Exactly the opposite is true. The farther to the right we go in figure 1, the *less* random variation occurs. The variable farthest to the right, the relationship of the car to its lane, can remain constant within a few inches for hour after hour. We find that this is the *most* stable variable in the chain, and that as we go backward up the chain toward Chip's nervous system, the random-looking variations get larger and larger. At the beginning of the chain the variations become totally unpredictable.

Consider figure 2; we added the effects of external events on a nervous system. According to the old picture still fundamental to most life sciences, external events act on the physical structure of the nervous system (along with internal events such as changes in body chemistry), and cause outputs to occur. Those outputs have consequences which show up at the end of the chain as behavioral patterns. To study the organization of behavior, you manipulate the external events, and look for regular behaviors that result (of course, you find them).

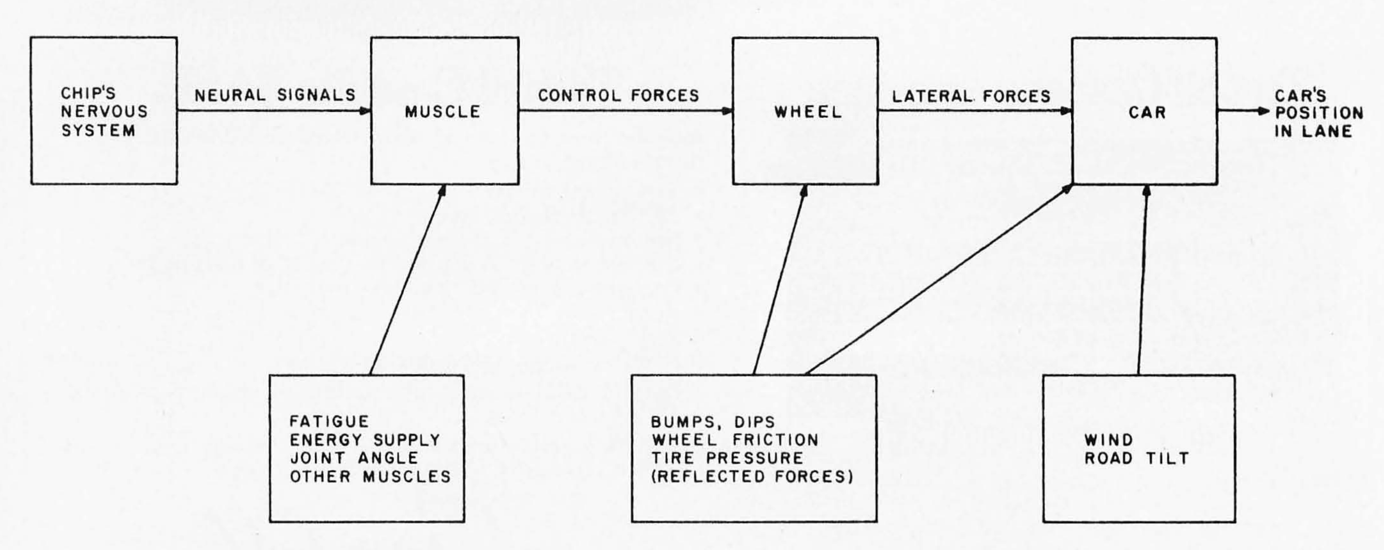
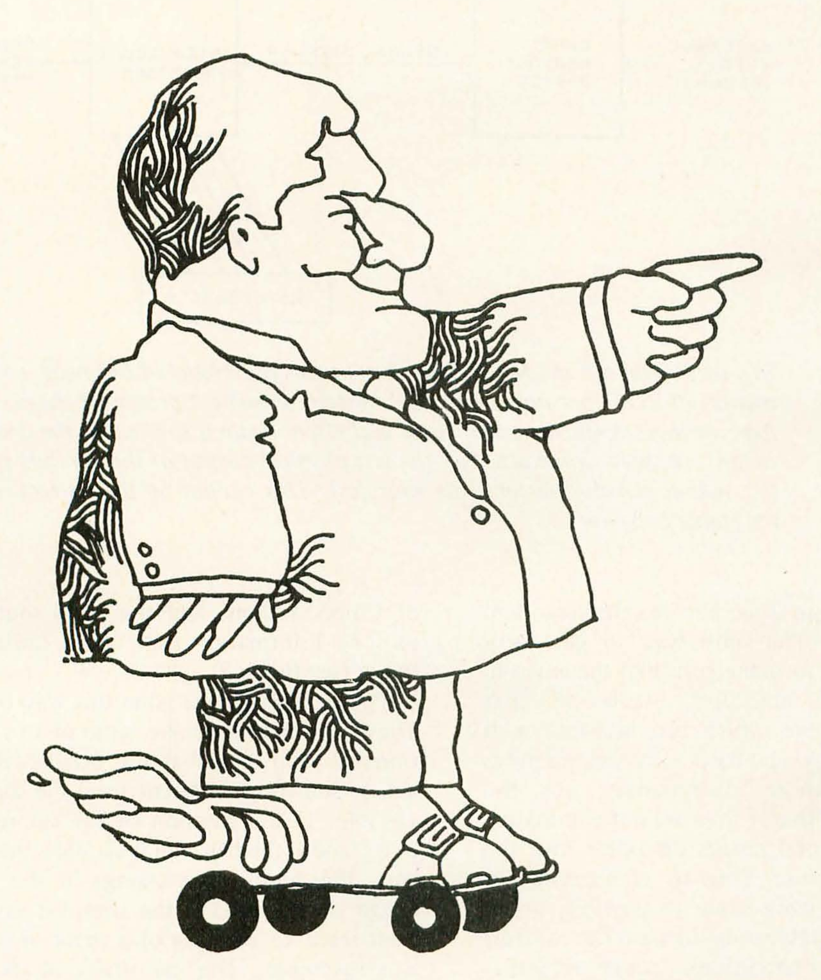


Figure 1: The cause and effect chain leading to behavior. The behavior called "driving in a straight line" is anything but simple. Some psychologists speak of behavior as simply being emitted by an organism, but this is clearly an inadequate concept. Between the nervous system and the stable pattern it appears to produce, disturbances come into play, having just as much effect on the final outcome as the nervous system has. Nevertheless, the most regularity appears at the end of this chain, and the least at the beginning.



HERE'S A QUICK POINT YOU SHOULD KNOW ABOUT

The fastest floating point BASIC for any micro.

TSC BASIC for the 6800 is the fastest floating point BASIC for ANY 8 bit microprocessor. No longer will the 6800 take a back seat to the 6502, 8080, or Z80! And with the TSC name, you know it's top quality.

TSC BASIC is not only fast, but complete with over 50 commands and functions. Features include six digit floating point math, full transcendental functions, unlimited string length, if/then/else construct, logical operators, and two-dimensional arrays including string arrays. The disk versions for FLEX™ 1.0 and 2.0 support random access data files (the mini FLEX™ version does not).

A cassette version requires 10K while the disk versions require at least 12K. No source listings included. With KCS cassette - $39.95; mini FLEX™ - $49.95; FLEX™ 2.0 - $54.95; and FLEX™ 1.0 - $59.95. Soon to come are a business BASIC and 6809 BASIC.

TSC

Technical Systems Consultants, Inc.

All orders should include 3% for postage and handling (8% on foreign orders). Send 25¢ for a complete software catalog.

Box 2574
W. Lafayette, IN 47906
(317) 463-2502

Circle 363 on inquiry card.

About the Author

William T Powers has been exploring the meaning of control theory for studies of human nature since 1953, when he was working as a health physicist at the University of Chicago. Since that time he has spent a number of years (to 1960) in medical physics, and then another 13 (to 1975) as Chief Systems Engineer for the Department of Astronomy at Northwestern University. His occupation has been designing electronic, optical, and mechanical systems for science. Powers' book, Behavior: The Control of Perception *(Aldine, 1973) was quite well received. At the moment he consults in one-of-a-kind electronics.*

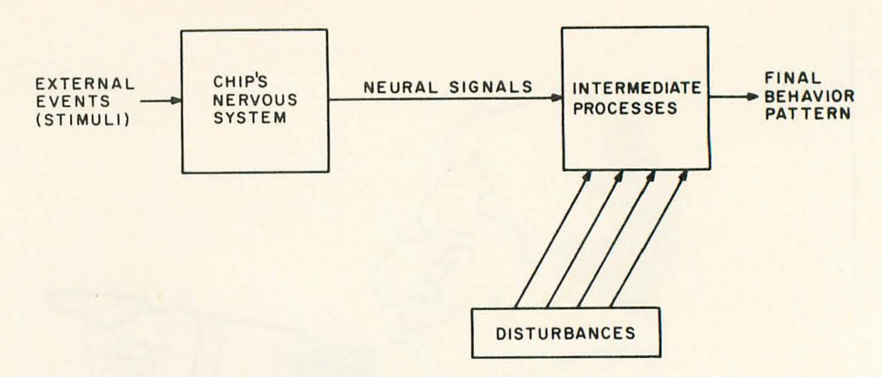


Figure 2: The old model of behavior. In this old model of behavior, environmental "forces" act on the nervous system to make it produce behavior. The logic of this straight-through, cause and effect chain is spoiled by the presence of disturbances which act after *the last physical output of the nervous system (ie: neural signals that activate muscles). This cannot be the correct model for stable behavior.*

But in figure 2 we also see those random disturbances. The only way to get away from them is to make sure that the environment remains absolutely stable (ie: that nothing happens which can interfere with behavior). The standard approach requires eliminating those disturbances, for the simple reason that if they are not eliminated, the experimental results disappear into the background noise. Thus by eliminating disturbances as completely as possible, under the guise of establishing standard (ie: control) experimental conditions, some scientists have swept this basic problem under the rug. They have also done away with the principal tool we have for understanding how these systems really work. If there are no disturbances, then the idea of a cause-effect chain running from external events through the organism to behavior seems to hold up, more or less. As soon as natural disturbances are allowed to occur, we find that the *overall* connection from external event to final behavior remains as clear as ever; but, the model of what happens in between falls to pieces with a loud crash.

Closing the Loop

There seems to be nothing wrong with figure 2; nothing, that is, except that it cannot account for the regularities of behavior. There is something wrong; something has been left out. Let's focus on the final variable in the chain, the position of the car relative to the lane. What variable that could affect Chip's senses, do you suppose, would have the most to do with his manipulations of the steering wheel? The position of the car relative to the lane. This variable is both the consequence of Chip's actions, and the main source of sensory information that could cause him to act (see figure 3).

Psychologists have gone this way before. They have tried to make sense of this situation by supposing that the behavioral variable is somehow different from the stimulus variable. If the position of the car relative to its lane is the behavioral variable, then perhaps the onset of a change in the visual image of the road is the stimulus variable. That leads to the idea of a *chain* of stimuli and responses. The car drifts in its lane; that stimulates Chip's nervous system to make a response, which affects the physical position of the car in its lane, which causes a new change in the stimulus, and so on around and around.

There are several severe difficulties with this explanation. In the first place, there is no way to separate the visual image from the position of the car; these are just two ways of talking about one whole physical situation in which a certain collection of interdependent variables changes simultaneously. The alternation between stimulus and response is completely imaginary, as anyone who drives knows. If causes and effects really were sequential, and chased themselves around and around the loop, it is unlikely that Chip would keep the car on the road for more than ten seconds. In part 2 we'll do a proper simulation in BASIC, and you will see that when the system is designed to behave sequentially, the result is most likely to be violent oscillations.

There is no reason at all to make an artificial distinction between the position of the car on the road as a behavioral response and as the stimulus which causes the response. Only one physical situation

MICRO WORLD

TOLL-FREE
1-800-528-1418

LOWEST PRICES • FAST DELIVERY • DEPENDABLE SERVICE

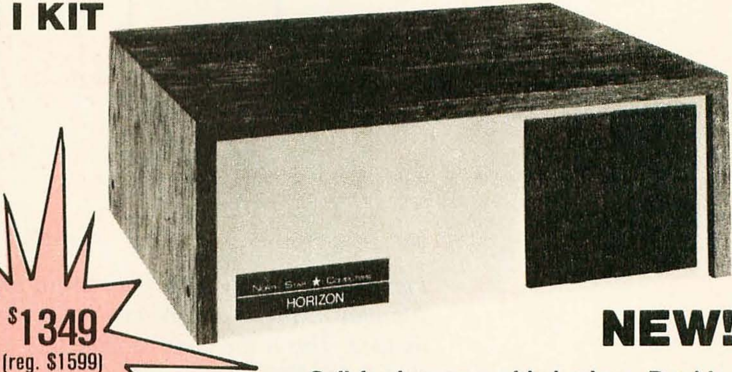
DOUBLE DENSITY HORIZON I KIT

double the storage at the same price!

Today's best buy. 180K bytes per disk. The chosen computer for two MicroWorld systems . . . Autoscribe — The Paperwork Manager™ and Bookkeeper — The Office Accountant™. Single density still runs on your new Horizon, or you can copy and convert all North Star software and programs to double density.

- exclusive application software
- add'l 16K memory (kit), $349
- add'l disk drive (kit), $349

$1349 (reg. $1599)



NEW!

Call for low assembled prices. Double density also available on North Star disk sub-system, $599 kit.

RUN SUPERIOR MICROPOLIS BASIC



MICROPOLIS DISK DRIVE FOR TRS-80

$499 Introductory Price (reg. $599)



HEATH DATA SYSTEMS PRINTER

$895

NEW LOW PRICE!

SOROC IQ 120 VIDEO TERMINAL



$795 (reg. $995)

Immediate Shipment!

TI 810 PRINTER



$1695 (reg. $1895)

MicroWorld Specials

Integral Data Printer	$ 749
Add for Tractors	$ 150
Mime Terminal	$ 742
Hazeltine 1410 Terminal .	$ 765
Hazeltine 1500 Terminal .	$1097
Teletype Model 43 Printer	$1019
Dataproducts M-200 Printer	$2921
Vector Graphics MZ	$3225
Exidy 16K Sorcerer	$1035
Data General microNOVA	$CALL
Qume Sprint 5 RO Printer (55cps)	$2895
DEC LS 120 Printer	$CALL
Centronics 779 Printer (incl. tractors)	$1095
Micropolis 1042 Mod I Drive	$CALL
TI 59 Calculator	$ 219
Novation CAT Modem ..	$ 199
TI 820 Terminal	$2199
North Star Horizon Software	$CALL
Dataproducts B-300 Printer	$5346
IPSI 1620 Diablo RO	$2705
Hitachi 9" Monitor	$ 184
Cromemco System III ...	$5299
Imsai VDP-80	$CALL

MICRO WORLD™

Bringing information technology to your doorstep

1425 W. 12th Pl. • Tempe, AZ 85281 • 602-894-1193

VISA No extra charge for credit card orders



Circle 233 on inquiry card.

Freight collect, F.O.B. Tempe
No other handling charges

Formerly Byte Shop Mail Order

CALL TOLL-FREE 1-800-528-1418

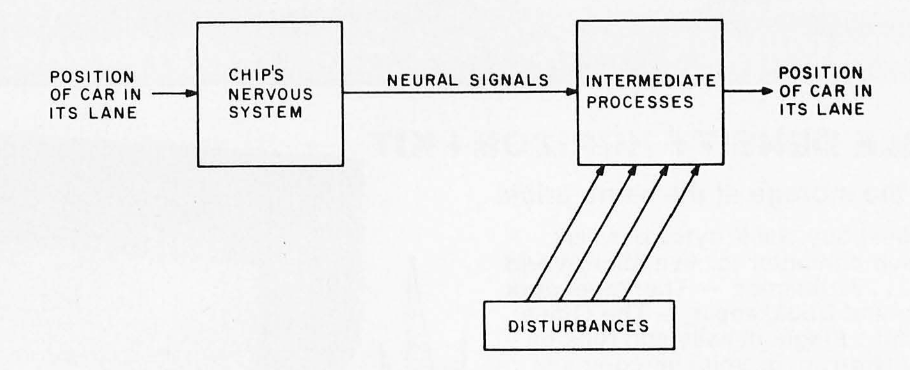


Figure 3: A slightly different view of the old model of behavior. The principle stimulus involved in driving a car in a straight line is the position of the car in its lane. This is the same variable that is the measure of behavior. The variable that is the final outcome of Chip's actions is the same variable that provides inputs to the nervous system that is acting. The variable at the causal end of the chain is the same variable found at the effective end of the chain.

exists, and there is no need to present it in two disguises. The position of the car on the road is both an effect of Chip's actions and the sensory situation which leads (with a little help from Chip) to those actions. There is a closed loop of cause and effect, and the position of the car is just one part of that loop.

Now we begin to draw a diagram of a proper control system. In figure 4, three physical quantities are shown, an *output* quantity, an *input* quantity, and a *disturbing* quantity.

The output quantity corresponds to an output of Chip's that is entirely due to himself (ie: perhaps due to the neural signals reaching his muscles or to some variable farther down the chain of figure 2, revealed when disturbances are known or can be legitimately eliminated).

The input quantity is the variable that is stabilized by the variations in Chip's output. Thus we call the input quantity, here, the position of the car relative to its lane. Of course, by that we mean whatever it is about that position that can be a sensory input to Chip (ie: probably a visual image of the hood of the car and the road beyond, framed in the windshield).

Between the output quantity and the input quantity is placed a *feedback function.* This function expresses the physical links that exist between Chip's output quantity and the input quantity. In the case of a moving car, if the output quantity were the angle of the steering wheel, which it might be if the angle is also a controlled quantity, then the effect of the wheel angle would be a continual change of car position, and the feedback function would have to include at least one time integration. The feedback function is simply a description of the physical processes which give each magnitude and direction of the output quantity a contribution to the state of the input quantity.

In figure 4 we also include disturbances as an integral part of the diagram of the system. The disturbing quantity in this case would be wind velocity and direction, and the *disturbance function* connecting it to the input quantity would express the way in which aerodynamic laws convert wind velocity into effects on the car's position in its lane.

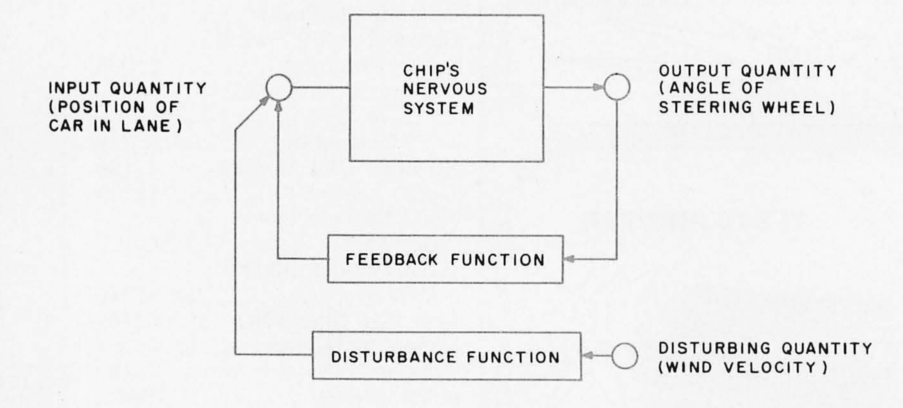


Figure 4: Closing the loop. By rearranging the relationships shown in figure 3 and eliminating the redundant appearance of the car position, we create a closed loop diagram. This is the general form of a control system diagram that will be used in this series from now on. The controlled variable is always the input quantity; the output quantity is the means of control. The single disturbance shown represents the net effective disturbance if more than one is acting at the same time. The disturbing function is chosen to provide the proper net contribution to the input quantity. The feedback function represents links external to the behaving nervous system through which outputs are transformed into contributions to the state of the input quantity.

The state of the input quantity, therefore, can be expressed in terms of all effects which contribute to it. We have shown only the output quantity and the disturbance due to wind. Many other disturbances — low tires, or tight wheel bearings, or gradation in the road — could also contribute to the state of the input quantity at the same time. All disturbances, however, can be reduced to a single one, since no matter what the cause of the disturbance, the only effect that matters is the effect on lateral position of the car.

Chip himself can be represented by a function, a function that converts the sensed position of the car into a steering wheel angle. This *system function* (system, being short for behaving system) will surely contain delays, nonlinearities, and even variations of its parameters. At first glance it may seem a terrible oversimplification to reduce a whole human being to a simple input/output box, but the situation isn't that bad. We are centering this diagram around the input quantity, not around Chip as a whole; therefore the "Chip box" does not wholly represent him, but only that part which reacts to changes in the input quantity by altering the output quantity. Furthermore, the Chip box (ie: the system function) is not quite as simple as it seems even after being simplified a great deal.

The functions connecting the variables in this closed loop can be extremely complex, and even to approach this system analytically will obviously require some approximations. This is not the place to justify every simplification; sometimes complex mathematics are required to reach a simple conclusion. I'll drop some hints along the way about how the simplified model is generated and why it works, but if you really want to get into this, study a text on servomechanism design.

Simulating Chip

Let us conclude by building a working simulator of Chip driving the car. This is just a hint of what this 4 part series of articles will develop. Building the simulator requires building some special numbers into the program without any explanation at present. The point is to enjoy the simulation, and get used to the idea that everything in a control loop happens at the same time.

We will assume that the steering wheel angle to left or right of center is Chip's output quantity, and that there are no disturbances that can interfere at this point. This output quantity will be called A.

Under the influence of A alone, the car would drift sideways at a rate proportional to A, for small deviations from the center of the lane. Designating the crosswind velocity as W, if W were the only influence acting, the car would drift sideways at a rate proportional to W (in this somewhat oversimplified universe). In the BASIC program we will assume that each iteration corresponds to a fixed amount of elapsed time, so the distance D that the car will drift during any one iteration is simply the sum of the two influences acting on it (line numbers correlate with listing 1):

```
7    D = K1 * W + K2 * A
```

The position, I, of the car relative to its lane will change by an amount D on each iteration:

```
8    I = I + D
```

Now I must introduce a detail: if we just had Chip respond proportionally to the deviation of car position, we would have to make his muscles so flabby that hardly any response would occur, unless we wanted to demonstrate self-immolating oscillations. We have to take care of two destabilizing factors. First, the feedback function is essentially

Look for Personal Software™ products at the dealer nearest you!

ALABAMA
BYTE SHOP
Huntsville, AL 35805
COMPUTERLAND
Huntsville, AL 35805
CPU, INC.
Montgomery, AL 36104
THE LOGIC STORE
Opelika, AL 36801
ALASKA
ALPHA ELECTRONICS
Anchorage, AK 99503
ARIZONA
MILLET'S TV & RADIO
Mesa, AZ 85204
PERSONAL COMPUTER PLACE
Mesa, AZ 85202
COMPUTERLAND OF PHOENIX
Phoenix, AZ 85016
COMPUTER SHOWROOM
Tucson, AZ 85710
ARKANSAS
COMPUTERLAND
Little Rock, AR 72212
DATACOPE
Little Rock, AR 72204
CALIFORNIA
JAY-KERN ELECTRONICS
Bakersfield, CA 93305
BYTE SHOP
Burbank, CA 91506
BYTE SHOP OF SACRAMENTO
Citrus Heights, CA 95610
COAST COMPUTER CENTER
Costa Mesa, CA 92627
CAPITOL COMPUTER SYSTEMS
Davis, CA 95616
COMPUTERLAND SAN DIEGO EAST
El Cajon, CA 92020
COMPUTERLAND OF EL CERRITO
El Cerrito, CA 94530
BUSINESS ENHANCEMENT
COMPUSERVICE
Escondido, CA 92027
CHANNEL DATA SYSTEMS
Goleta, CA 93017
RAINBOW COMPUTING
Granada Hills, CA 91344
JADE COMPUTER PRODUCTS
Hawthorne, CA 90250
BYTE SHOP OF HAYWARD
Hayward, CA 94541
COMPUTERLAND OF HAYWARD
Hayward, CA 94541
COMPUTERLAND OF WEST L.A.
Inglewood, CA 90302
COMPUTER COMPONENTS
OF SOUTH BAY
Lawndale, CA 90260
COMPUTERLAND OF SOUTH BAY
Lawndale, CA 90260
A-VIDD ELECTRONICS
Long Beach, CA 90815
COMPUTERLAND
Los Altos, CA 94022
BYTE SHOP
Mountain View, CA 94040
HOBBY WORLD ELECTRONICS
Northridge, CA 91324
COMPUTERS-MADE-EASY
Palmdale, CA 93550
BYTE SHOP OF PLACENTIA
Placentia, CA 92670
COMPUTER CENTER
Riverside, CA 92503
CAPITOL COMPUTER SYSTEMS
Sacramento, CA 95821
COMPUTERLAND
San Bernadino, CA 92404
COMPUTERLAND OF SAN DIEGO
San Diego, CA 92111
COMPUTER MERCHANT
San Diego, CA 92115
COMPUTERLAND OF
SAN FRANCISCO
San Francisco, CA 94105
VIDEO GAMES & COMPUTERS
San Francisco, CA 94118
COMPUTERLAND OF SAN JOSE
San Jose, CA 95129
COMPUTERLAND (Central)
San Leandro, CA 94577
BYTE SHOP
San Louis Obispo, CA 93401
MARIN COMPUTER CENTER
San Rafael, CA 94903
ADVANCED COMPUTER PRODUCTS
Santa Ana, CA 92705
BYTE SHOP
Santa Clara, CA 95051
COMPUTER FORUM
Santa Fe Springs, CA 90670
THE COMPUTER STORE
Santa Monica, CA 90401
SANTA ROSA COMPUTER CENTER
Santa Rosa, CA 95404
BYTE SHOP
Suisun, CA 94585
COMPUTERS PLUS
Sunnyvale, CA 94087
BYTE SHOP OF TARZANA
Tarzana, CA 91356
COMPUTERLAND OF
THOUSAND OAKS
Thousand Oaks, CA 91360
SMALL SYSTEM SOFTWARE
Thousand Oaks, CA 91360
COMPUTER COMPONENTS
Van Nuys, CA 91411
COMPUTERLAND
Walnut Creek, CA 94598
BYTE SHOP
Westminster, CA 92683
COMPUTER COMPONENTS OF
ORANGE COUNTY
Westminster, CA 92683
COLORADO
BYTE SHOP
Boulder, CO 80301
COMPUTERLAND
Colorado Springs, CO 80917
AMPTEC
Denver, CO 80216
COMPUTERLAND
Denver, CO 80222
BYTE SHOP
Englewood, CO 80110
MICRO WORLD ELECTRONIX
Lakewood, CO 80226
CONNECTICUT
COMPUTERLAND OF FAIRFIELD
Fairfield, CT 06430
JRV COMPUTER STORE
Hamden, CT 06518
THE COMPUTER STORE
Hartford, CT 06103
THE COMPUTER STORE
Windsor Locks, CT 06096
WASHINGTON D.C.
COMPUTER CABLEVISION
Washington, D.C. 20007
FLORIDA
COMPUTERLAND
Boca Raton, FL 33432
THE COMPUTER STORE
Bradenton, FL 33505
THE COMPUTER STORE
Clearwater, FL 33516
UCATAN
Destin, FL 32541
BYTE SHOP
Fort Lauderdale, FL 33334
COMPUTERLAND
Fort Lauderdale, FL 33308
COMPUTERS FOR YOU
Fort Lauderdale, FL 33312
DATA MOVERS
Fort Meyers, FL 33901
FOCUS SCIENTIFIC ENTERPRISES
Miami, FL 33132
GRICE ELECTRONICS
Pensacola, FL 32589
COMPUTER AGE
Pompano Beach, FL 33062
PAPERBACK BOOKSMITH
Sarasota, FL 33581
AMF ELECTRONICS
Tampa, FL 33612
MICRO COMPUTER SYSTEMS
Tampa, FL 33609
COMPUTER CENTER OF
PALM BEACHES
West Palm Beach, FL 33409
GEORGIA
ADVANCE COMPUTER TECHNOLOGIES
Atlanta, GA 30328
COMPUSHOP
Atlanta, GA 30342
DATAMART
Atlanta, GA 30305
THE LOGIC STORE
Columbus, GA 31906
COMPUTERLAND OF ATLANTA
Smyrna, GA 30080
HAWAII
COMPUTERLAND
Honolulu, HI 96813
MICROCOMPUTER SYSTEMS
Honolulu, HI 96813
RADIO SHACK (Dealer)
Lihue, HI 96766
IDAHO
NORTHWEST COMPUTER CENTER
Boise, ID 83704
ILLINOIS
COMPUTERLAND OF
ARLINGTON HEIGHTS
Arlington Heights, IL 03904
FARNSWORTH COMPUTER CENTER
Aurora, IL 60505
KAPPEL'S COMPUTER STORE
Belleville, IL 62220
DOW-COM
Carbondale, IL 62901
BYTE SHOP
Champaign, IL 61820
THE ELEKTRIK KEYBOARD
Chicago, IL 60614
EMMANUEL B. GARCIA JR.
AND ASSOCIATES
Chicago, IL 60613
PERSONAL COMPUTER
Chicago, IL 61820
COMPUTERLAND
Downers Grove, IL 60515
COMPUTER STATION
Granite City, IL 62040
ORCUTT BUSINESS MACHINES
La Salle, IL 61301
ILLINI MICROCOMPUTERS
Naperville, IL 60540
COMPUTERLAND OF NILES
Niles, IL 60648
COMPUTERLAND
Oak Lawn, IL 60453
COMPUTERLAND OF PEORIA
Peoria, IL 61614
WALLACE ELECTRONICS
Peoria, IL 61614
DATA DOMAIN
Schaumburg, IL 60195
INDIANA
DATA DOMAIN OF FORT WAYNE
Fort Wayne, IN 46805
HOME COMPUTER CENTER
Indianapolis, IN 46220
PUBLIC COMPUTING
Lafayette, IN 47904
IOWA
SYNCHRONIZED SYSTEMS
Des Moines, IA 50310
THE COMPUTER CENTER
Waterloo, IA 50701
KANSAS
THE COMPUTER ROOM
Overland Park, KS 66212
PERSONAL COMPUTER CENTER
Overland Park, KS 66206
COMPUTER SYSTEMS DESIGN
Wichita, KS 67214
LOUISIANA
COMPUTER SHOPPE
Metaire, LA 70002
MARYLAND
COMPUTERLAND
Rockville, MD 20855
COMPUTER WORKSHOP
Rockville, MD 20852
COMPUTERS ETC.
Towson, MD 21204
COMPUTERS UNLIMITED
Towson, MD 21204
MASSACHUSETTS
THE COMPUTER STORE
Burlington, MA 01803
THE COMPUTER STORE
Cambridge, MA 02139
CPU SHOP
Charlestown, MA 02129
MAD HATTER SOFTWARE
Dracut, MA 01826
NEW ENGLAND ELECTRONICS
Needham, MA 02194
NEW ENGLAND ELECTRONICS
Springfield, MA 01103
MICHIGAN
NEWMAN COMPUTER EXCHANGE
Ann Arbor, MI 48104
NEW DIMENSIONS IN COMPUTING
East Lansing, MI 48823
COMPUTER HOUSE DIV.
Jackson, MI 49202
COMPUTERLAND OF
GRAND RAPIDS
Kentwood, MI 49508
COMPUTRONIX
Midland, MI 48640
COMPUTER MART OF ROYAL OAK
Royal Oak, MI 48073
TRI CITY COMPUTER MART
Saginaw, MI 48603
COMPUTERLAND
Southfield, MI 48034
LEVEL FOUR PRODUCTIONS
Westland, MI 48185
MINNESOTA
COMPUTERLAND
Bloomington, MN 55431
MINN. MICRO SYSTEMS
Minneapolis, MN 55454
MISSISSIPPI
OXFORD SOFTWARE CO.
Oxford, MS 38655
MISSOURI
FORSYTHE COMPUTERS
Clayton, MO 63105
COMPUTER COUNTRY
Florissant, MO 63031
GREATEST GRAPHICS
Springfield, MO 65804
NEBRASKA
OMAHA COMPUTER STORE
Omaha, NE 68127
NEVADA
HOME COMPUTERS
Las Vegas, NV 89109
NEW HAMPSHIRE
TRS-80 SOFTWARE EXCHANGE
Milford, NH 03055
COMPUTERLAND OF NASHUA
Nashua, NH 03060
BITS, INC.
Peterborough, NH 03458
NEW JERSEY
COMPUTER LAB OF NJ
Budd Lake, NJ 07828
COMPUTER EMPORIUM
Cherry Hill, NJ 08002
COMPUTER MART OF NJ
Iselin, NJ 08830
MSM ELECTRONICS
Medford, NJ 08055
COMPUTERLAND
Morristown, NJ 07960
COMPUTERLAND
Paramus, NJ 07652
COMPUTER NOOK
Pine Brook, NJ 07058
COMPUTER CORNER
Pompton Lakes, NJ 07442
COMPUTER ENCOUNTER
Princeton, NJ 08540
TYPTRONIC COMPUTER STORE
Ramsey, NJ 07446
NEW YORK
COMPUTERLAND
Buffalo, NY 14150
COMPUTERLAND
Carle Place, NY 11514
COMPUTER SHOP OF SYRACUSE
De Witt, NY 13214
THE COMPUTER TREE
Endwell, NY 13760
LONG ISLAND COMPUTER
GENERAL STORE
Lynbrook, NY 11563
COMPUTER MICROSYSTEMS
Manhasset, NY 11030
COMPUTER SHOPPE
Middle Island, NY 11953
THE COMPUTER FACTORY
New York, NY 10017
COMPUTER MART OF NEW YORK
New York, NY 10016
DATEL SYSTEMS
New York, NY 10036
AUTOMATIC SYSTEMS
Poughkeepsie, NY 12603
COMPUTER HOUSE
Rochester, NY 14609
THE COMPUTER STORE
Rochester, NY 14618
HOME COMPUTER CENTER
Rochester, NY 14607
THE COMPUTER CORNER
White Plains, NY 10601
READOUT COMPUTER STORE
Williamsville, NY 14221
NORTH CAROLINA
BYTE SHOP
Charlotte, NC 28212
COMPUTERLAND
Charlotte, NC 28205
FUTUREWORLD
Durham, NC 27707
BYTE SHOP
Greensboro, NC 27401
MICROCOMPUTER SERVICES
Hickory, NC 28601
BYTE SHOP OF RALEIGH
Raleigh, NC 27605
OHIO
BASIC COMPUTER SHOP
Akron, OH 44314
CINCINNATI COMPUTER STORE
Cincinnati, OH 45246
21ST CENTURY SHOP
Cincinnati, OH 45202
DIGITAL DESIGN
Cincinnati, OH 45202
CYBER SHOP
Columbus, OH 43227
MICRO MINI COMPUTER WORLD
Columbus, OH 43213
COMPUTER SOLUTIONS
Dayton, OH 45409
DAYTON COMPUTER MART
Dayton, OH 45409
ASTRO VIDEO ELECTRONICS
Lancaster, OH 43130
COMPUTERLAND OF CLEVELAND
Mayfield Heights, OH 44121
RADIO SHACK (Dealer)
St. Clairsville, OH 43950
OKLAHOMA
HIGH TECHNOLOGY
Oklahoma City, OK 73106
MICROLITHICS
Oklahoma City, OK 73127
HIGH TECHNOLOGY
Tulsa, OK 74129
OREGON
THE COMPUTER STORE
Corvallis, OR 97330
CAMERA AND COMPUTER
EMPORIUM
Portland, OR 97205
COMPUTERLAND OF PORTLAND
Tigart, OR 97223
PENNSYLVANIA
BYTE SHOP
Bryn Mawr, PA 19010
PERSONAL COMPUTER CENTER
Frazer, PA 19355
COMPUTER AID
Latrobe, PA 15650
THE COMPUTER WORKSHOP
Murrysville, PA 15668
A B COMPUTERS
Perkasie, PA 18944
MICROTRONIX
Philadelphia, PA 19106
SOUTH CAROLINA
DATA MART
Greenville, SC 29607
TENNESSEE
MICROCOMPUTER STORE
Knoxville, TN 37919
COMPUTER LABS OF MEMPHIS
Memphis, TN 38117
DOC'S COMPUTER SHOP
Nashville, TN 37211
TEXAS
COMPUTERLAND OF AUSTIN
Austin, TX 78757
COMPUTERS 'N THINGS
Austin, TX 78731
MICRO COMPUTER SHOPPE
Corpus Christi, TX 78411
COMPUSHOP
Dallas, TX 75243
COMPUTERLAND
Dallas, TX 75231
KA ELECTRONICS SALES
Dallas, TX 75247
COMPUTER TERMINAL
El Paso, TX 79901
RAM MICRO SYSTEMS
Fort Worth, TX 76116
COMPUTERCRAFT
Houston, TX 77063
COMPUTERLAND OF HOUSTON BAY
Houston, TX 77058
NEIGHBORHOOD COMPUTER
Lubbock, TX 79401
COMPUTER PATCH OF SANTA FE
Odessa, TX 79762
COMPUSHOP
Richardson, TX 75080
THE COMPUTER SHOP
San Antonio, TX 78216
COMPUTER SOLUTIONS
San Antonio, TX 78229
WICHITA COMPUTER SYSTEMS
Wichita Falls, TX 76301
UTAH
ADP SYSTEMS
Logan, UT 84321
COMPUTER CONCEPTS GROUP
Salt Lake City, UT 84109
THE HI-FI SHOP
Salt Lake City, UT 84117
VERMONT
COMPUTERMART
Essex Junction, VT 05452
VIRGINIA
COMPUTER HARDWARE STORE
Alexandria, VA 22314
COMPUTERS PLUS
Alexandria, VA 22304
COW, INC.
Blacksburg, VA 24060
HOME COMPUTER CENTER
Newport News, VA 23606
COMPUTER TECHNIQUES
Richmond, VA 23235
THE COMPUTER PLACE
Roanoke, VA 24015
COMPUTER WORKSHOP
Springfield, VA 22151
COMPUTERLAND
Vienna, VA 22180
HOME COMPUTER CENTER
Virginia Beach, VA 23452
WASHINGTON
OMEGA NORTHWEST
Bellevue, WA 98004
COMPUTERLAND OF SOUTH
KING COUNTY
Federal Way, WA 98003
YE OLDE COMPUTER SHOPPE
Richland, WA 99352
THE COMPUTER SHOPPE
Seattle, WA 98115
EMPIRE ELECTRONICS
Seattle, WA 98166
PERSONAL COMPUTERS
Spokane, WA 99202
COMPUTERLAND
Tacoma, WA 98499
WISCONSIN
BYTE SHOP OF MILWAUKEE
Greenfield, WI 53227
COMPUTERLAND
Madison, WI 53711
MADISON COMPUTER STORE
Madison, WI 53711
COMPUTERLAND
Milwaukee, WI 53222
FOX VALLEY
COMPUTER STORE
Neenha, WI 54956
WYOMING
COMPUTER CONCEPTS
Cheyenne, WY 82001
AUSTRALIA
ELECTRONIC CONCEPTS PTY. LTD.
COMPUTERLAND
Sydney, N.S.W.
CANADA
COMPUSHOP
Calgary, Alberta T2N 2A4
THE COMPUTER SHOP
Calgary, Alberta T2T 4T9
ORTHON COMPUTERS
Edmonton, Alberta T5N 3N3
TJB MICROSYSTEMS
Edmonton, Alberta T5M 0H9
CONTI ELECTRONICS
Vancouver, B.C. V5W 2Z4
COMPUTER CITY
Winnepeg, Manitoba R3P 0H8
COMPUTERLAND
Winnepeg, Manitoba R3G 0M8
INTERACTIVE COMPUTER SYSTEMS
Frederickton, New Brunswick
MINICOMP SYSTEMS
Halifax, Nova Scotia B3K 2G1
KOBETEK SYSTEMS
Wolfville, Nova Scotia B0P 1X0
COMPUTERLAND
Burlington, Ontario
LYNTRONICS
Downsview, Ontario M2J 2W6
COMPUTER CIRCUITS
London, Ontario N6A 3H2
COMPUMART
Ottawa, Ontario K2A 1J2
COMPUTER INNOVATIONS
Ottawa, Ontario K1B 4A8
RICHVALE TELECOMMUNICATIONS
Richmond Hill, Ontario
THE COMPUTER CENTRE
Sarnia, Ontario N7T 1B4
COMPUTER MART
Toronto, Ontario M4G 3B5
THE COMPUTER PLACE
Toronto, Ontario M5V 1Z1
COMPUTER SPECIALIST
Toronto, Ontario M3K 1E7
HOME COMPUTER CENTRE
Toronto, Ontario M2M 3W2
HOUSE OF COMPUTERS
Toronto, Ontario
MARKETRON
Toronto, Ontario
MICRO-WARE
Toronto, Ontario M4E 2L2
COMPUCENTRE
Montreal, Quebec H1J 1Z4
FUTUR BYTE
Montreal, Quebec H3B 3C9
CUSTOM COMPUTING SERVICES
Saskatoon, Saskatchewan S7K 2B5
DIGITAL SERVICE
Saskatoon, Saskatchewan S7J 3A9
PUERTO RICO
MICROCOMPUTER STORE
Rio Piedras, PR 00921
ENGLAND
BYTE SHOP
Ilford, Essex
INFOGUIDE
London
CYTEK
Manchester M4 3E4
PETSOFT
Newbury, Berkshire RG13 1PB
KEEN COMPUTERS
Nottingham NG7 1FN
T & V JOHNSON
MICROCOMPUTERS
Camberly, Surrey
PETALECT
Woking, Surrey
J&J ELECTRONICS
Bexhill-on-Sea, East Sussex
OPTRONICS
Twickenham TW1 4RY
DENMARK
MICRO SYS
Copenhagen V1BYJ
FRANCE
SVEA
Paris 75008
GERMANY
BECK COMPUTERS
8 Munchen 70
ING. W. HOFACKER
8 Munchen 75
HOLLAND
COMPUTRON
Den Haag 2502 ER
ITALY
HOMIC MICROCOMPUTERS SRL
20123 Milano
SCOTLAND
MICRO CENTRE
Edinburgh EH3 5AA
SINGAPORE
THE COMPUTER CENTRE PTE. LTD.
Singapore 7
SWEDEN
MICROFUTURE
Stockholm 10322
SEMIDAKO
Uppsala 75353
SWITZERLAND
INTERFACE TECHNIC
Basel
INGENIEUBUREAU
Basel 1
DIALOG COMPUTER
Lucerne
ELBATEX
Wettingen

P.S.: VISICALC—How did you ever do without it?

Circle 302 on inquiry card.

```
 1 INPUT "WIND, MPH: ",W
 2 PRINT "WHEEL ANGLE, DEGREES",
 3 PRINT TAB(25),"CAR DEVIATION, FEET"
 4 FOR J=1 TO 10
 5 PRINT %7F1,"     ",A*10,
 6 PRINT %7F1,TAB(25),I
 7 D=.05*W+A
 8 I=I+D
 9 A1=-2*(I+.8*D)
10 A=A+.200*(A1-A)
11 NEXT
12 GOTO 1
13 END

RUN
      WIND, MPH: 20
      WHEEL ANGLE, DEGREES                      CAR DEVIATION, FEET
                 .0                                      .0
               -7.2                                     1.0
              -11.8                                     1.3
              -13.3                                     1.1
              -12.7                                      .8
              -11.3                                      .5
              -10.1                                      .4
               -9.5                                      .4
               -9.4                                      .4
               -9.6                                      .5
      WIND, MPH: -30
      WHEEL ANGLE, DEGREES                      CAR DEVIATION, FEET
               -9.8                                      .5
                8.0                                    -2.0
               19.3                                    -2.7
               23.1                                    -2.2
               21.6                                    -1.4
               18.3                                     -.8
               15.4                                     -.4
               13.8                                     -.4
               13.6                                     -.5
               14.0                                     -.7
      WIND, MPH: 40
      WHEEL ANGLE, DEGREES                      CAR DEVIATION, FEET
               14.6                                     -.8
              -10.1                                     2.7
              -26.0                                     3.7
              -31.2                                     3.1
              -29.2                                     2.0
              -24.6                                     1.0
              --20.5                                     .6
              -18.4                                      .5
              -18.0                                      .7
              -18.6                                      .9
      WIND, MPH: -50
      WHEEL ANGLE, DEGREES                      CAR DEVIATION, FEET
              -19.5                                     1.0
               12.3                                    -3.4
               32.6                                    -4.7
               39.3                                    -3.9
               36.8                                    -2.5
               30.9                                    -1.3
               25.7                                     -.7
               22.9                                     -.6
               22.4                                     -.9
               23.2                                    -1.1
```

Listing 1: A rough simulation of Chip driving the car in a straight line. Each iteration is assumed to correspond to a fixed time interval. Therefore, the distance the car drifts away from straight line travel is the sum of the wind and steering wheel angle. The simulation shows Chip trying to arrive at the wheel angle which will counteract the force of the blowing wind. If you repetitively use the same wind value, you will see that a steady wheel angle is arrived at. [I found it interesting that this simulation seems to settle down within 60 time units to a consistent value. Even changing wind values from +1000 to −1000 units was compensated for within 60 time units. . . . RGAC*]*

an integrator, and so puts a lag into the control process. This alone would not cause a problem, but Chip also contains a *transport lag;* he cannot actually produce an output at the same instant that the input occurs, nor can our program since it is evaluating equations one at a time. The integration lag we take care of by adding to the position I (which Chip senses) the variable D, which is approximately the first derivative of the input quantity. He senses the input quantity with some emphasis on its rate of change, which is actually a realistic model of human perception. This part of the stabilizing of the control action is done in step 9:

$$9 \quad A1 = K3 * (I + 0.8 * D)$$

We have computed a variable A1, the angle which the wheel would assume if Chip reacted instantly. But to handle the transport lag, we must slow his reponse, letting only a fraction K5 (between 0 and 1) of it occur during any one iteration. That is what step 10 does:

$$10 \quad A = A + K5 * (A1 - A)$$

This slowing technique will be used in the larger simulator next time. To see how it works, set A1 to 10.00, K5 to 0.25, and A to 0, and then simply keep doing step 10 with pencil and paper. A will gradually approach the value of A1 from any starting point.

The program in listing 1 asks for a wind velocity, and then proceeds to do ten iterations of the control loop, printing wheel angle A and car position deviation I each time. A positive number means the wind is blowing, the wheel is cocked, or the car has moved to the *right.* If you want to follow the program for more than ten iterations, give it the same wind again. It always starts where it left off.

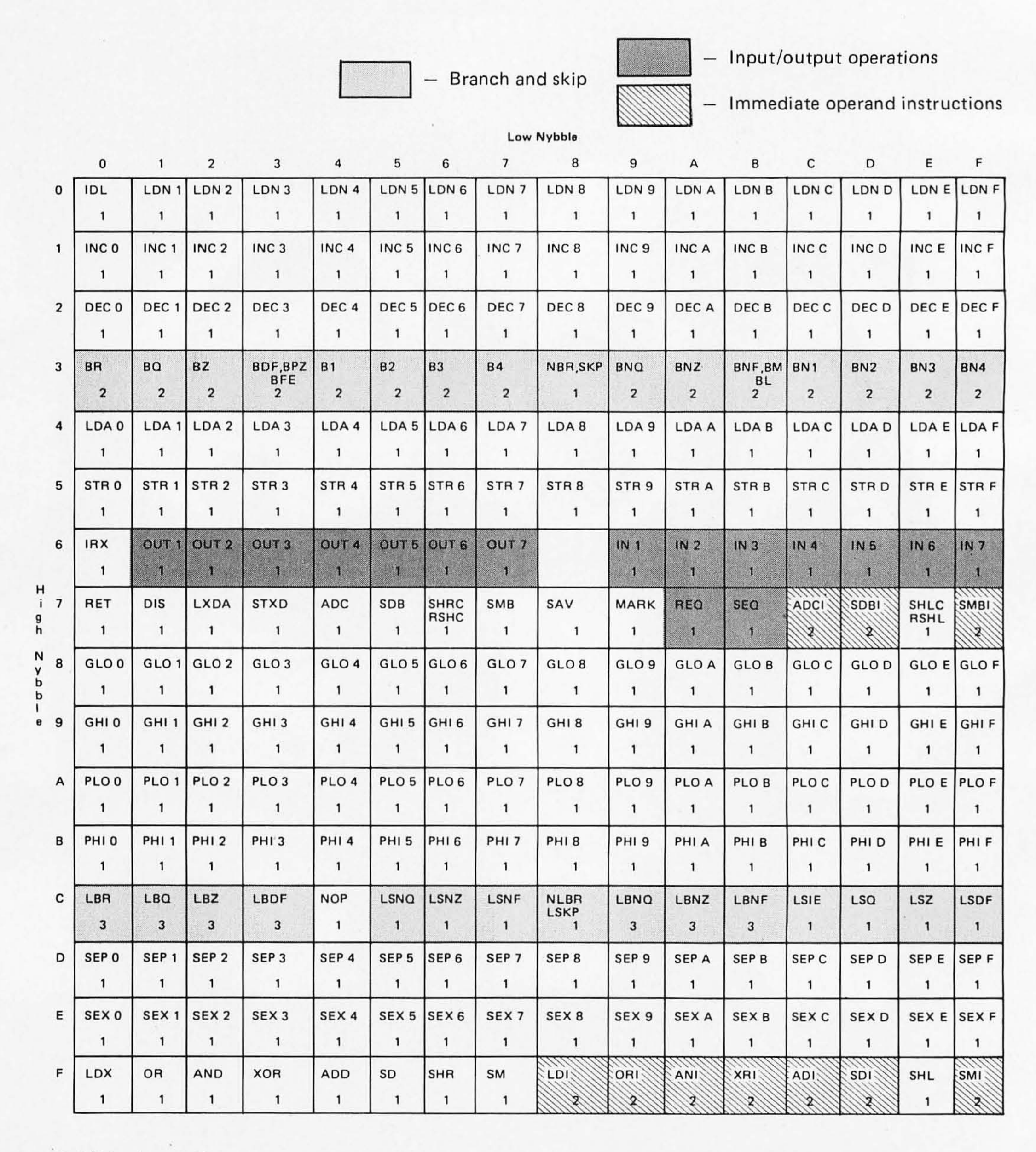
In part 2, we will begin exploring a model of the kind described in figure 4 and start the somewhat mind boggling task of retraining the intuition to think in closed loop terms instead of straight through cause and effect. There is a *big* difference. We'll see that, in general, control systems control what they sense, not what they do. We'll discover something called a *reference signal,* which functions in a control system exactly the way an inner purpose has always been supposed to function. In part 2, we'll see how perception figures into control. And we'll start working with a more extended BASIC simulator than the tiny one in listing 1. Parts of this simulator will be suitable for building into the computer part of a robot, should anyone want to carry matters that far.■

The 1802 Op Codes

Henry Melton
2511 Dovemeadow Dr
Austin TX 78744

The RCA/Hughes 1802 is an 8 bit microprocessor with a small but growing following. Its ease of interface and low CMOS power requirements make it attractive for many small applications. The accompanying chart of op codes for the 1802 illustrates all of the 255 variations. I have used the RCA mnemonics.

The 1802 is organized around sixteen 16 bit registers. These can be used as program counters, index registers, subroutine pointers, and general data storage registers. The interrupt and direct memory access features of the 1802 also make use of specific registers for their operation. There is an 8 bit accumulator (D), and three 1 bit flags: DF for the carry flag, IE for the interrupt enable flag, and Q for the direct output flag. There are also four 4 bit registers: two to hold the current op code, one



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	IDL 1	LDN 1 1	LDN 2 1	LDN 3 1	LDN 4 1	LDN 5 1	LDN 6 1	LDN 7 1	LDN 8 1	LDN 9 1	LDN A 1	LDN B 1	LDN C 1	LDN D 1	LDN E 1	LDN F 1
1	INC 0 1	INC 1 1	INC 2 1	INC 3 1	INC 4 1	INC 5 1	INC 6 1	INC 7 1	INC 8 1	INC 9 1	INC A 1	INC B 1	INC C 1	INC D 1	INC E 1	INC F 1
2	DEC 0 1	DEC 1 1	DEC 2 1	DEC 3 1	DEC 4 1	DEC 5 1	DEC 6 1	DEC 7 1	DEC 8 1	DEC 9 1	DEC A 1	DEC B 1	DEC C 1	DEC D 1	DEC E 1	DEC F 1
3	BR 2	BQ 2	BZ 2	BDF,BPZ BFE 2	B1 2	B2 2	B3 2	B4 2	NBR,SKP 1	BNQ 2	BNZ 2	BNF,BM BL 2	BN1 2	BN2 2	BN3 2	BN4 2
4	LDA 0 1	LDA 1 1	LDA 2 1	LDA 3 1	LDA 4 1	LDA 5 1	LDA 6 1	LDA 7 1	LDA 8 1	LDA 9 1	LDA A 1	LDA B 1	LDA C 1	LDA D 1	LDA E 1	LDA F 1
5	STR 0 1	STR 1 1	STR 2 1	STR 3 1	STR 4 1	STR 5 1	STR 6 1	STR 7 1	STR 8 1	STR 9 1	STR A 1	STR B 1	STR C 1	STR D 1	STR E 1	STR F 1
6	IRX 1	OUT 1 1	OUT 2 1	OUT 3 1	OUT 4 1	OUT 5 1	OUT 6 1	OUT 7 1		IN 1 1	IN 2 1	IN 3 1	IN 4 1	IN 5 1	IN 6 1	IN 7 1
7	RET 1	DIS 1	LXDA 1	STXD 1	ADC 1	SDB 1	SHRC RSHC 1	SMB 1	SAV 1	MARK 1	REQ 1	SEQ 1	ADCI 2	SDBI 2	SHLC RSHL 1	SMBI 2
8	GLO 0 1	GLO 1 1	GLO 2 1	GLO 3 1	GLO 4 1	GLO 5 1	GLO 6 1	GLO 7 1	GLO 8 1	GLO 9 1	GLO A 1	GLO B 1	GLO C 1	GLO D 1	GLO E 1	GLO F 1
9	GHI 0 1	GHI 1 1	GHI 2 1	GHI 3 1	GHI 4 1	GHI 5 1	GHI 6 1	GHI 7 1	GHI 8 1	GHI 9 1	GHI A 1	GHI B 1	GHI C 1	GHI D 1	GHI E 1	GHI F 1
A	PLO 0 1	PLO 1 1	PLO 2 1	PLO 3 1	PLO 4 1	PLO 5 1	PLO 6 1	PLO 7 1	PLO 8 1	PLO 9 1	PLO A 1	PLO B 1	PLO C 1	PLO D 1	PLO E 1	PLO F 1
B	PHI 0 1	PHI 1 1	PHI 2 1	PHI 3 1	PHI 4 1	PHI 5 1	PHI 6 1	PHI 7 1	PHI 8 1	PHI 9 1	PHI A 1	PHI B 1	PHI C 1	PHI D 1	PHI E 1	PHI F 1
C	LBR 3	LBQ 3	LBZ 3	LBDF 3	NOP 1	LSNQ 1	LSNZ 1	LSNF 1	NLBR LSKP 1	LBNQ 3	LBNZ 3	LBNF 3	LSIE 1	LSQ 1	LSZ 1	LSDF 1
D	SEP 0 1	SEP 1 1	SEP 2 1	SEP 3 1	SEP 4 1	SEP 5 1	SEP 6 1	SEP 7 1	SEP 8 1	SEP 9 1	SEP A 1	SEP B 1	SEP C 1	SEP D 1	SEP E 1	SEP F 1
E	SEX 0 1	SEX 1 1	SEX 2 1	SEX 3 1	SEX 4 1	SEX 5 1	SEX 6 1	SEX 7 1	SEX 8 1	SEX 9 1	SEX A 1	SEX B 1	SEX C 1	SEX D 1	SEX E 1	SEX F 1
F	LDX 1	OR 1	AND 1	XOR 1	ADD 1	SD 1	SHR 1	SM 1	LDI 2	ORI 2	ANI 2	XRI 2	ADI 2	SDI 2	SHL 1	SMI 2

Table 1: RCA/Hughes 1802 instruction set. The op codes and the number of bytes used by each are shown.

to designate which of the R registers is the program counter, and one to designate which of the R registers is the index pointer. Lastly, there are four EF external pins that are sensed as conditional branch flags.

11 commands make up the bulk of the 1802 op codes. Each is 1 byte in length, with the first nybble designating the operation, and the last nybble acting as a 4 bit immediate operand. They account for 175 of the op codes (LDN does not operate on register 0) and allow data transfer to and from the R indexed memory, incrementing and decrementing the R registers, and setting the P and X registers. There are four major operations (hexadecimal 80 through BF) that move bytes of data between the accumulator and the two halves of the 16 bit registers.

The arithmetic and logical operations use the accumulator with either the immediate data or the indexed memory contents as the other operand.

There are three types of branches. The short branch uses the immediately following byte as the next address in the local 256 block of memory space. This is *not* a relative jump; the immediate data just replaces the low byte of the program counter. The long branch uses the two following bytes, high byte first, to construct the branching address. The skip instructions skip over following instructions if the tested condition is true. The short skip skips only one byte, but all the others skip over two bytes of code. Notice that the short branch instructions include the external flags as possible testable conditions, allowing quick direct serial input.

There is also a set of I/O (input/output) instructions to transfer a byte to or from one of seven possible I/O ports. The external Q pin can be set or reset to give quick direct serial output.

The SAV, MARK, RET, and DIS instructions can be used to implement recoverable interrupt and nested subroutines by using a stack in memory.

The last instructions are NOP and IDL. The IDL instruction places the machine on hold until an interrupt or direct memory access request occurs.

The 1802 is a well-designed computer package, and CMOS is *the* technique in battery powered applications.■

The History of Computing
The IBM 7070

IBM 7070s undergoing final checkout before shipment.



An IBM 7070 operator's console.



A typical IBM 7070 configuration. The engineering console (for repair diagnostic information) is in the table in the foreground.

Keith S Reid-Green
Member of Technical Staff
RCA, David Sarnoff Research Center
Princeton NJ 08540

The second computer generation is said to have begun with the advent of the transistor. An equally important advance was the realization that most programmers would soon be programming not in symbolic machine language but in machine independent high level languages. This led hardware designers to build instructions that simplified compiling of programs, such as editing, table lookup and string scanning instructions. Some of these instructions are found in the IBM 7070, announced in 1959.

The 7070, like the 650, was built around a 10 digit signed decimal word. Signs and digits were stored in a two-out-of-five code; five bits were used to represent a digit, of which exactly two were "on". The five bits were assigned values of 0,1,2,3 and 6, so a little fudging had to be followed in order to represent 0. Since 3 can be represented two ways, 0-3 or 1-2, the former was used as the value for 3 and the latter for 0. Alphabetic characters were represented by digit pairs; hence, only five letters could be stored in a word. To distinguish between alphabetic and numeric representation, three signs were allowed: +, −, and @, represented by the codes for 9, 6, and 3, respectively.

Two memory sizes, 5,000 and 9,990 words, were available. The larger size could have been 10,000 words, but addresses 9991-9999 referred to addressable registers, including three accumulators. The small memory size was seen as a drawback, but a remarkably powerful instruction set tended to offset this problem. The 7070 might have been the start of a new decimal based family of computers if IBM had added a memory paging feature to allow for expansion.

The instruction format used the sign and first two digits as an operation code, two digits for an indexing address, two digits for field definition or instruction augmentation, and four digits for an address. Hence, 200 different instruction types and 99 index words were possible. In fact, 190 instructions were implemented, and memory locations 0001 through 0099 could be used as index "registers." Field definition allowed arithmetic to be performed on parts of words. If field definition 58 was specified in an ADD instruction, the contents of digit positions 5 thru 8 of the addressed value would be added to the specified accumulator.

I/O (input/output) units did not look much different from first generation equipment, but in fact embodied several significant improvements. Magnetic tape units could read and write in low (200 bits per inch) or high density (556 bits per inch). Data channels allowed overlap of data transfer and processing, and a priority I/O scheme allowed for simultaneous peripheral operation on line—spooling, as it is still called. The idea behind spooling is that punched cards may be read and lines printed during times when the machine is simultaneously doing something else. The main program gets and puts records on a faster medium, such as a disk, and is therefore not forced to wait for the relatively slow card reader or printer.

File handling capability on the 7070 was very sophisticated due to a feature called *block transmission*. A record could be read into several noncontiguous blocks of memory under control of RDWs (record definition words). The programmer specified in a *tape read* instruction the unit number of the tape to be read and the address of the first of a sequence of one or more record definition words. Each definition contained the starting and ending address of a block (set of consecutive words) into which data was to be read. Each record definition word in the sequence had a positive sign except the last one, which was negative. If the physical record ended before all definitions were exhausted, the remaining were ignored. If the last record definition word was exhausted before the end of the record was reached, the remaining data were not transferred into memory. When a record was written, record definition words were similarly used to gather data from various parts of memory without actually transferring them into one contiguous memory block.

Variable length records could also be handled easily through use of a *tape read per record mark control* instruction. A record mark was a special character having the keypunch code 0-2-8, which was written as a not equal sign (≠). When the tape was read under record mark control, the normal sequence of data transmission to storage via record definition words was followed until a record mark was read. This caused the 7070 to cease transmission of data to the block specified by the current definition and to move on to the next.

For example, suppose a tape record contained:

ABCDEFGHI≠JKLM≠NOPQR

and a tape read per record mark control

instruction referred to the record definition words:

+0010001005	words 1000 thru 1005
+0014601480	words 1460 thru 1480
−0019992999	words 1999 thru 2999,

ABCDE would be read into location 1000, FGHI into 1001, JKLM into 1460 and NOPQR into 1999.

A major problem arose because of the special nature of alphabetic data. In first generation computers, conventions had been established regarding coding of alphabetic characters on tape, and these conventions were carried on into the 7070. Some means of differentiating between alphabetic and numeric data had to be established, however, since the old-fashioned "overpunched" numeric sign used in unit-record equipment and early computers could not always be distinguished from alphabetic characters. This problem was resolved on the 7070 by assuming that a tape was alphabetic until a delta character was read, whereupon the delta was not transferred into memory, but served instead to change the mode of data transfer to numeric. The next delta flipped the mode back to alphabetic, and so on. A *delta* on the seven track BCD (binary coded decimal) tape (in which the tracks were labeled CBA8421) consisted of the CB8421 bits. Thus the three words:

+0123443210
−5678998765
@7461796368

would be written on tape as △01234-432105678998765△MARCH. When reread into a 7070, a perfect translation took place back into the three words shown above. However, when read into any other computer, the delta characters transferred into memory, giving IBM 1401 programmers considerable annoyance. Nevertheless, it is safe to say that the 7070 represented a very large step between first generation and modern computers. Most apparent among the second generation refinements was the recognition that computers did not have to be classified as commercial or scientific machines but in fact could be used to solve problems in both of the disciplines. On the one hand, the record definition word concept made sorting extremely fast, while hardware implemented floating point instructions were available for calculation. ■

INFO 2000 DISK SYSTEMS:
A LOT LESS THAN YOU EXPECT.

Less Cost

The DISCOMEM Controller board costs us less to manufacture. So your complete INFO 2000 Disk System costs you less – at least $400 less than comperable disk systems.

Less Hardware

Only three S-100 boards are needed to create a complete, high-performance disk-based microcomputer system–the DISCOMEM Controller Board, a 32K memory board, and any 8080, 8085 or Z80 CPU board. You don't need extra interface or EPROM boards since DISCOMEM contains 2 serial ports, 3 parallel ports and provision for 7K of EPROM and 1K of RAM.

LessTime

The INFO 2000 Disk System is incredibly fast! Using the PerSci Drives with voice coil positioning, disk seek times are up to 8 times faster than with other drives. A full disk-to-disk copy and verification takes well under a minute. Formatting and verifying a new diskette takes less than half a minute. Reloading CP/M† from diskette takes a fraction of a second.

Less Space

The system is remarkably compact, requiring only 1/2 to 1/3 the space taken by other 2-drive disk systems.



Less Hassle

The INFO 2000 Disk System eliminates the "I/O configuration blues" by incorporating all necessary interface ports. A CP/M Loader and all I/O drivers are contained in EPROM so there is no need for special software customization. Just plug the system into your S-100 microcomputer and begin immediate operation using the CP/M disk operating system. The INFO 2000 Disk System is supported by the most extensive library of software available, including 3 different BASICs, 2 ANSI FORTRAN IVs, several assemblers, text editors, debugging tools, utilities and numerous applications packages.

Less Errors

This disk system uses full size 8" diskettes and standard IBM 3740 recording format. So you're assured superior protection against errors, and full interchangeability with other CP/M-based systems.

It all adds up to more capability for your money.

The complete INFO 2000 Disk System comes completely assembled and tested. It includes dual diskette drives, the DISCOMEM Controller, power supply, cabinet, cables and the CP/M disk operating system––everything you need for immediate plug-in-and-go operation with your microcomputer. This means less time, hassle, hardware, space, errors and less money than for comparable equipment. Now, isn't that a lot less than you'd expected in a dual disk system?

INFO 2000 Disk Systems are also available for Digital Group and Heath H8 microcomputers. Dealer inquiries welcomed.

NOW IN STOCK: DOUBLE-SIDED DRIVES! Doubles your storage capacity*

INFO 2000
CORPORATION
20620 South Leapwood Avenue
Carson, California 90746
(213) 532-1702

† CP/M is a registered trademark of Digital Research.

*Extra cost option.

Circle 173 on inquiry card.

Artificial Intelligence and Entropy

R M Kiehn
Physics Dept
University of Houston
Houston TX 77004

Although this article is written with a sense of science fiction creativity and speculation, the concepts involved are based on sound ideas that recently won for Ilya Prigogine the 1977 Nobel prize (see reference 1). No longer can it be said that the laws of thermodynamics prohibit the creation of artificial intelligence.

Computer intelligence has long been a hazy dream in many enthusiasts' eyes, but so often that gleam has been glazed over by a rational response from the scientific community: "You can get out of any computer only what you put in. The computer, after all, has to be told what to do. It can't think."

These valid scientific arguments are based, essentially, on the laws of thermodynamics, especially the second law, concerning entropy. The second law says in effect that for all (isolated) systems at or near equilibrium, any process will have a tendency to increase the system's entropy.

Shannon has developed the idea that entropy is related to the inverse of information (see reference 2); so as entropy goes up, information goes down. The second law of thermodynamics seems to demand that this decrease in information shall always happen for mechanical or electrical machines.

The conclusion that machines will never be capable of thinking seems irrefutable.

But man is a biological machine, and yet he thinks. How can these two seemingly contradictory ideas be reconciled with the second law? One of the greatest scientific puzzles of the last century was that the idea of a thinking biological machine *cannot* be brought into accord with the laws of *equilibrium* thermodynamics! The paradox stands.

The brilliance of I Prigogine lay in his recognition that another regime far from local thermodynamic equilibrium may exist for complex interacting systems. Exactly what this new regime is all about took many years of study to formulate and understand, but in the last two years the verification of this new thermodynamic concept has been achieved by a small but convincing number of chemical experiments (see reference 3). Simply, the new thermodynamic regime is a regime *far* from equilibrium where semi-steady state modes of collective interactions can self-organize in complex systems. In this collective mode of self-organization, far from thermodynamic equilibrium, the entropy of the system drops dramatically; the information content rises. The increasing entropy dictum of the second law is no longer valid because the system is not in equilibrium.

The key features of systems that can support such entropy-decreasing modes are that:

1. They are not isolated systems: they draw energy and material from their surroundings: that is, they feed.
2. They are complex systems of *many* interacting parts, and the interactions must be nonlinear.

These conditions seem to be necessary but not sufficient. A complex system that has these properties need not self-organize into an entropy decreasing mode, but if these conditions are met, the low entropy modes are now scientifically recognized possibilities.

Theoretical models of complex chemical systems have indicated that the minimum number of subparts that will form a collective mode far from equilibrium is measured in the thousands. This result is striking; it could have been that it would take statistically large mole numbers (10^{23}) of parts to form self-organized systems, but no, it appears that thousands, or a few tens of thousands, of subsystems interacting nonlinearily can, for certain ranges of parameters, flip into self-organized modes of decreasing entropy.

For the first time, thermodynamics (albeit of the nonequilibrium variety) can be used to substantiate entropy decreasing modes, biological systems, and thinking machines! Using these ideas as a basis, it is conceivable that a collection of subsystem computers linked together nonlinearly could, for certain ranges of coupling parameters, self-organize into a collective mode of operation in which the entropy of the complex system decreases.

The gleam in the young experimenter's eye need not be glazed over and extinguished by a thermodynamic "it's impossible" argument. The loophole in the law has been discovered.

A few years ago such a project would have been economically impossible, but now with the advent of cheap small volume microprocessors it is conceivable that, say, ten thousand computer subsystems could be assembled and linked together in one complex. Moreover, following the theoretical advances pioneered by I Prigogine, such a complex might be arranged to operate in a collective mode far from equilibrium in which the entropy content drops dramatically, the information content skyrockets, and the idea of artificial intelligence need no longer be squashed by thermodynamic dictum. It indeed may be possible to create a thinking machine.

REFERENCES

1. C Nicolis and I Prigogine, *Self-Organization in Non-Equilibrium Systems,* John Wiley & Sons, New York, 1977.
2. C E Shannon and W Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, 1949.
3. J S Turner, *Journal of Physical Chemistry* (to be published).
4. H Haken, *Synergetics,* Springer-Verlag, Berlin, 1977.

Appendix: Collective Mode Systems of Low Entropy

After I submitted the above article to BYTE, editor Carl Helmers asked if I could provide a tutorial example explaining the ideas of a collective mode system with low entropy. Perhaps the most striking example of such a system is given by the laser (see reference 4).

The typical laser consists of an optical cavity, a set of atoms – such as He-Ne– connected to the outside world by means of a power supply. Energy flows continually into (and out of) the system so it cannot be considered to be isolated. If the system gain parameter is below a critical value, the gas atoms may be excited by the energy source and subsequently release their excess excitation energy through the emission of photons. The photons all have random phase, ie: the emission of photons by the various atoms in the gas is random, and not coherent. The system has a high entropy content associated with the disorder and the random emission of photons. The system does reach a steady state, *not* a collective mode state, and *not* a state of equilibrium, because it continually receives energy across its boundary.

If the gain parameter of the system is slowly increased to above the critical value, a dramatic transition takes place – far from equilibrium. The system lases, ie: the gas atoms no longer emit the photons in a random manner, but instead they de-excite in a coherent collective mode. The emission of one photon from one atom stimulates the emission of another photon from another atom, and so on. The photons are emitted with the same phase; disorder has been removed, and the system entropy drops dramatically.

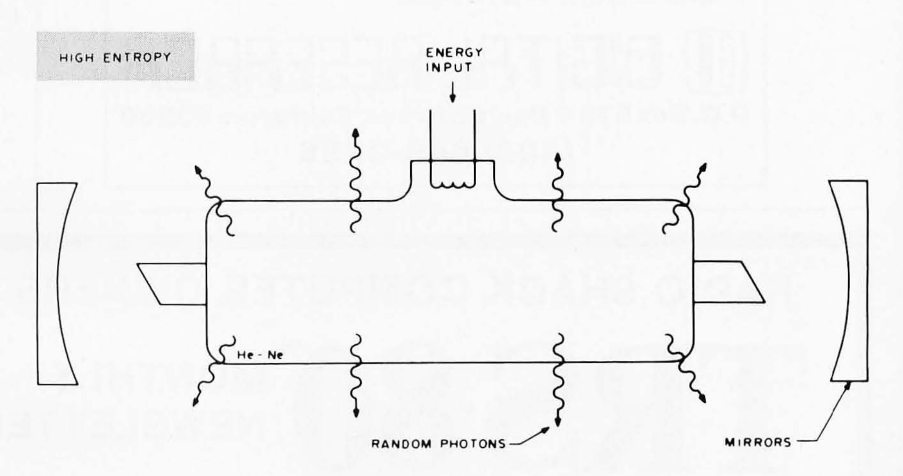


Figure 1: The He-Ne discharge glows in a disordered random fashion. The emission of photons is random, disordered and associated with the notion of high entropy.

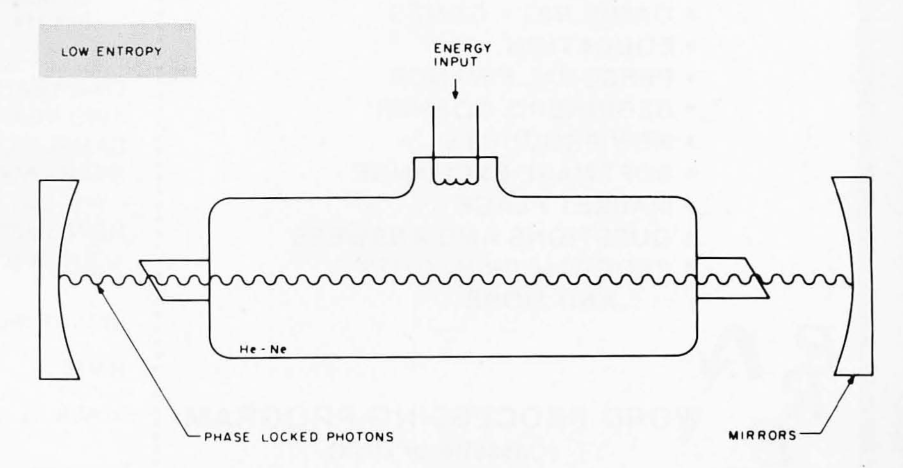


Figure 2. For the system "gain" above the critical value lasing action takes place in the He-Ne gas. The radiation is phase locked into a collective mode associated with low entropy.

Circle 95 on inquiry card.

The notions of disorder and randomness are concepts associated with the notion of entropy. The decrease of the laser's entropy is due to lack of disorder in the lasing state.

These ideas are described in figures 1 and 2. Figure 1 represents a laser being supplied with energy from the power source, but with the gain of the cavity below the critical value. The photon emission is random and the system has high entropy.

On the other hand in figure 2 the gain factor of the system is above the critical value and the system has flipped into a low entropy collective mode emitting highly ordered radiation. The atoms *radiate collectively*.

The almost incredible feature of many complex systems interacting nonlinearly is that they may *self-organize* into these low entropy modes. In particular, biological systems appear to be of this type. Numerous examples of collective mode systems appear in the physical world, but most of them occur at low temperature; super conductors and super fluids are examples. The self-organization into collective modes far from equilibrium at modest to high temperatures is yet another idea.■

Circle 159 on inquiry card.

INTRODUCING THE .4 MEGABYTE DUAL DRIVE MINIFLOPPY FOR 8K PETS!



NOW WITH 400K (.4 MEGABYTES) OF DISK STORAGE

CUSTOMER ORDERS ARE NOW BEING SHIPPED WITHIN TEN DAYS!!!

DKH641 IS A PRODUCT OF COMPUTHINK.

- DUAL MINI FLOPPY DRIVE WITH 100K PER DISK SIDE FOR TOTAL 200K ON LINE.
- COMMERCIAL LOADING SPEEDS * 8K LOADS IN 2.6 SECONDS COMPLETE.
- DOS REORGANIZES SPACE AFTER SAVE OR ERASE FOR EFFICIENT STORAGE.
- DISKMON ADDS OVER 20 COMMANDS TO BASIC INCLUDING DISK DATA FILES.
- DISKMON COMMANDS SUPPORT COMMERCIAL PRINTER OFF PARALLEL PORT SUCH AS CENTRONICS LINE OF PRINTERS (AVAILABLE FROM NEECO)
- FULL DISK SOFTWARE SUPPORT * FORTRAN & PLM COMPILERS *
- 90 DAY MANUFACTURER'S WARRANTY ON HARDWARE * READY TO USE ON DELIVERY. FULL MANUAL AND UTILITY DISKETTE INCLUDED.
- CALL OR WRITE FOR COMPLETE INFORMATION ON THE "DISKDRIVER".
- THIS SYSTEM REQUIRES EXPANDAPET MEMORY (MINIMUM 16K)

DKH642	— DUAL DRIVE SYSTEM, COMPLETE WITH DISKMON	$1295.00
ASM789D	— PET ASSEMBLER ON DISKETTE WITH MANUAL	$49.95
*LNK456	— AUTOLINK LINKING LOADER ON DISKETTE WITH MANUAL	$49.95
*PLM400	— COMPILER ON DISKETTE (AVAILABLE JUNE)	$250.00
DKL067	— DISKMON ASSEMBLER LISTING/DOS	$19.95
DATA100	— COMPLETE DATA BASE SYSTEM (PRICE APPROXIMATE)	$400.00
BASCOMP	— BASIC COMPILER ON DISKETTE	
NGP200	— 20 GAMES ON DISKETTES	$49.95
BKGAM	— BACKGAMMON ON DISKETTE	$24.95
MICRO	— MICROCHESS ON DISKETTE	$24.95
CEN779(1)	— CENTRONICS 779-1, ROLL FEED DOT MATRIX COMMERCIAL PRINTER	$1245.00
CEN779(2)	— SAME AS 779(1) BUT WITH TRACTOR FEED — PLUG INTO PET	$1345.00
AX10M	— ELECTROSTATIC 5.5 INCH PRINTER WITH ALL OF PET'S GRAPHICS	$495.00

PET COMPUTER

NEECO NOW OFFERS A FULL ONE YEAR WARRANTY ON ALL PETS! — AN ADDITIONAL 9 MONTHS!



by COMMODORE

WHY NOT BUY FROM THE BEST?

8K PET	$ 795
24K PET (8 + 16K)	$1195
32K PET (8 + 24K)	$1295

THE ABOVE PRICES INCLUDE THE INSTALLATION AND TESTING OF INTERNAL MEMORY BOARDS.

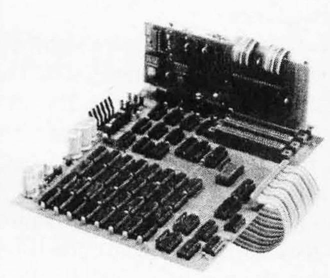
- ★ NEECO IS A CUSTOMER ORIENTED, FULL SERVICE COMPANY.
- ★ PETS RECEIVE 48 HR. 'BURNIN' BY NEECO BEFORE SHIPMENT.
- ★ FULL CUSTOMER SERVICE AND FULL PRODUCT SUPPORT.
- ★ 48 HR MAXIMUM 'TURNAROUND' ON PET WARRANTY SERVICE ON PETS PURCHASED FROM NEECO.
- ★ FULL PRE-PURCHASE INFO AVAILABLE FROM OUR PET INFO PACKAGE — WE ANSWER CUSTOMER QUESTIONS!
- ★ AUTOMATIC SOFTWARE/HARDWARE UPDATES VIA OUR PET OWNERS MAILING LIST — CALL/WRITE TO BE LISTED!
- ★ COMMERCIAL QUANTITIES AVAILABLE.
- ★ WE ALSO MARKET REPLACEMENT RAMS & ROMS, ETC.
- ★ OFF THE SHELF DELIVERIES (NO DEPOSIT REQUIRED).

SCHOOL INQUIRIES INVITED!

INTERNAL MEMORY EXPANSION FOR PET!

EXPANDAPET™

INTERNAL MEMORY EXPANSION UNIT FOR 8K PETS



DEALER INQUIRIES INVITED

- MOUNTS EASILY INSIDE YOUR PET
- EASY TO INSTALL (15 MINUTES)
- NO DEGRADATION OF PET SYSTEM
- USES LOW POWER DYNAMIC RAMS
- 90 DAY PART&LABOR, 1 YR-RAMS.
- 30 DAY MONEY BACK GUARANTEE.
- MOUNTING SLOTS FOR 4 BOARDS.
- CALL/WRITE FOR ADDITIONAL INFO
- DEALER INQUIRES INVITED.

EXPANDAPET PRICES

16K (+8K PET = 24K)	$425
24K (+8K PET = 32K)	$525
32K (+8K PET = 40K)	$615

OPTIONAL PLUG-IN BOARDS
4K EPROM DAUGHTER BOARD USING 2716 EPROMS $50

32K UNIT ALLOWS 8K OF ASSEMBLY LANGUAGE SUBROUTINES ACCESSED VIA THE USR COMMAND.

EXPANDAPET IS A PRODUCT OF COMPUTHINK.

FULLSIZED TYPEWRITER KEYBOARD FOR PET!



- COMMERCIAL QUALITY KEYBOARD WITH METAL ENCLOSURE.
- BASIC TYPEWRITER DESIGN FOR TOUCHTYPISTS.
- SINGLE KEY FUNCTIONS FOR ALL CURSOR CONTROLS. SHIFT/RUN, INSERT, CLEAR SCREEN/HOME CURSOR, MORE.
- FUNCTIONS SIMULTANEOUSLY WITH PET'S KEYBOARD.
- PLUGS DIRECTLY INTO PET'S LOGIC BOARD.
- DOES NOT USE USER OR IEEE-488 PORTS.
- NPK-101 IS FULLY TESTED & READY TO USE.
- ATTACHES DIRECTLY TO FRONT OF PET'S FRAME.
- CAN BE USED AS A REMOTE TERMINAL (SPECIAL ORDER).
- 30 DAYS TRIAL PERIOD * 90 DAY WARRANTY.
- CALL OR WRITE FOR FULL SPECS

NOW AVAILABLE! $139.95

NPK-101 IS A PRODUCT OF NEW ENGLAND ELECTRONICS.

OUR NEW JUNE PET CATALOG IS NOW AVAILABLE.
CALL OR WRITE FOR OUR *FREE* PET PRODUCTS CATALOG.

DOMESTIC & OVERSEAS DEALER INQUIRIES INVITED ON * MEMORY * KEYBOARD * FLOPPY

NEECO

NEW ENGLAND ELECTRONICS CO., INC.
679 HIGHLAND AVE., NEEDHAM, MASS. 02194
MON. - FRI. 9:30 - 5:30, EST.

(617) 449-1760
MASTERCHARGE OR VISA ACCEPTED
TELEX NUMBER 951021, NEECO

BASIC Text Editor

Fred R Ruckdeschel
773 John Glenn Blvd
Webster NY 14580

Like many an individual who uses computers at work and for personal purposes, I have a tendency to write. At work, this tendency is expressed in the usual ways as memos, project oriented documentation and more general papers. In personal life, this tendency is expressed by the existence of this article and several others which preceded it.

After some considerable chicken scratching it became apparent that my method for collecting ideas and creating a final manuscript was not optimum. The problem became clearly defined when one day I looked up from my note pad only to see my IMSAI arrogantly sleeping while I worked.

The literary approach I had been using was reasonably conventional. First in the procedure was the generation of an outline which contained ideas organized by section. Then a handwritten rough draft was composed in which these thoughts were structured within each section and expanded upon. The draft was then reviewed and changes made until legibility was threatened. If the writing had been for "work" the next step would have been to have the draft retyped. However, as writing for microcomputer journals is a private pursuit, a paid typist is used sparingly.

The next step was therefore to go back through the draft and reedit, being very careful with es and is, and then to give it to the typist for final preparation. The typist's job was to take the grubby and somewhat illegible pile of paper and transform it into a nicely typed manuscript suitable for submission. However, the last step was invariably one of carefully applying correction fluid, scissors and tape in the proper proportions to assemble a truly final version. Usually the alterations were due to my own errors, though my excellent typist often adds a few variations as part of her own editing contribution. As a result, I have never sent an original to an editor, always a copy which hid the horrors of the manuscript's creation. After all, what editor wants flakes of dried correction fluid sprinkled on his or her clothes and desk?

A few further considerations convinced me that it was time to make a change. First, I cannot stand reading my own handwriting. Second, I can type much faster than I can write. Third, it was a convenient time to design an editing system. The reason for the latter was that I had a video display, IMSAI and North Star disk combination at home and a very nice Diablo printer, Altair 8800B and North Star disk setup available at work. Thus there was basic compatibility between the two systems via the floppy diskette. I had taken care to assure that the two systems were software compatible, particularly with respect to IO.

The general idea was to use my personal system to compose and edit versions of a manuscript and then to bring the finished form, on diskette, to *work to be printed out* on the Diablo. In practice it turned out to be more convenient to have listings of the various versions to work from. This made the iterations much easier. There is no replacement for hard copy when writing either a program or an article.

One of the goals established for the eventual editor software was that it should not be unique to my particular hardware configuration. Rather it was to be translatable to other systems with a minimum of change. For this reason BASIC was chosen as the implementation language, though machine language would have led to a much better utilization of memory as well as higher speed. The particular BASIC used was North Star, Version 6, Release 3. This interpreter has string manipulation functions which are very convenient for developing editing routines. These functions can also be translated into counterparts a la Microsoft BASIC. This will be discussed more later.

Text editors naturally require significant computer storage capabilities. The one shown in listing 1 is no exception. For those who have disk based systems other than North Star, the conversion of this program for use with another system would be through changes in the disk access subroutines which are clearly defined on the listing. For those who do not have disk hardware, but rather cassette IO, storage and retrieval can also be accomplished through the disk access subroutines after the appropriate modifications are made.

The program also has the ability to automatically save and retrieve the text from active memory. This is presently not in effect in the version shown in listing 1, but can be brought to life by removing statements 2000 and 2100. This feature is useful if the computer is a little unstable and has a tendency to crash; at least the latest text version might be saved in a protected memory region if the program did not get as far as saving the text on diskette or tape. The

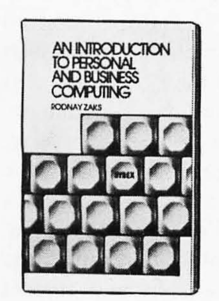


THE LEADER IN MICROCOMPUTER EDUCATION

PRESENTS

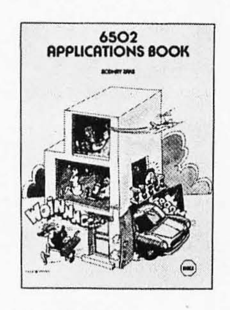
BEST-SELLING BOOKS

Used by Universities and Industry worldwide (ten languages)

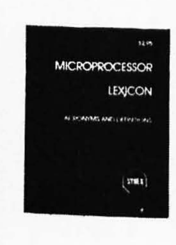












C200 - AN INTRODUCTION TO PERSONAL AND BUSINESS COMPUTING
Rodnay Zaks, 250 pp
A comprehensive introduction to small computers, their peripherals, and what to select. **$6.95**

C201 - MICROPROCESSORS: FROM CHIPS TO SYSTEMS
Rodnay Zaks, 416 pp
The basic text on all aspects of microprocessors and the assembly of a system. **$9.95**

C202 - PROGRAMMING THE 6502
Rodnay Zaks, 250 pp
A complete introductory programming text for the 6502. **$10.95**

D302 - 6502 APPLICATIONS BOOK
Rodnay Zaks, 200 pp
Connecting a 6502 board to the outside world: from home alarm to music and industrial control. **$12.95**

C207 - MICROPROCESSOR INTERFACING TECHNIQUES
A. Lesea & R. Zaks, 416 pp
How to interface a microprocessor to external devices: from keyboard to ADC to floppy disk, including standard busses. **$11.95**

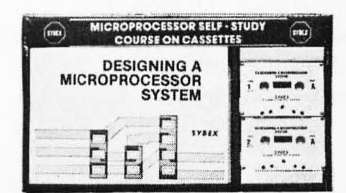
X1 - MICROPROCESSOR LEXICON, 125pp
All the definitions of the microprocessor world in a convenient pocketbook format. **$2.95**

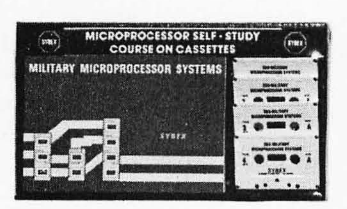
AVAILABLE AT BOOKSTORES, COMPUTER STORES, AND ELECTRONIC SHOPS EVERYWHERE!

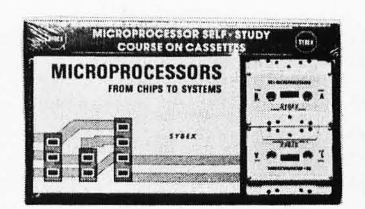
SELF-STUDY COURSES ON CASSETTES

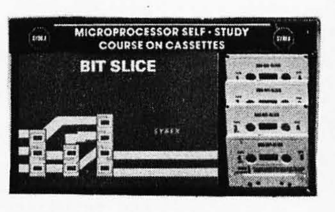
"The time-efficient way to learn"™

Self-study courses include two to eight audio-cassettes and a special book. They offer the fastest way to learn the topic covered (from ½ to two days). Highly effective.









1 - INTRODUCTORY - SHORT (2.5 hrs ea)
- S1-INTRODUCTION TO MICROPROCESSORS $29.95
- S2-PROGRAMMING MICROPROCESSORS $29.95
- S3-DESIGNING A MICROPROCESSOR SYSTEM $29.95
- S10B-INTRODUCTION TO PERSONAL AND BUSINESS COMPUTING $21.90

2 - INTRODUCTORY - COMPREHENSIVE (10 to 12 hrs ea)
- SB1-MICROPROCESSORS $59.95
- SB2-MICROCOMPUTER PROGRAMMING $59.95

3 - SPECIALIZED (4.5 to 6 hrs ea)
- SB3-MILITARY MICROPROCESSOR SYSTEMS $49.95
- SB5-BIT-SLICE $49.95
- SB6-INDUSTRIAL MICROPROCESSOR SYSTEMS $49.95
- SB7-MICROPROCESSOR INTERFACING $49.95

TO ORDER

By Phone: 415 848-8233, Visa, M.C., Amer Express.
By mail: Include payment.
Shipping: add $1.50 per book (UPS) or 65¢ (4th class - allow 4 weeks). Double for cassettes and overseas.
Tax: In California add tax.

FREE DETAILED CATALOGUE



Dept B6
2020 Milvia Street
Berkeley, CA 94704
Tel 415 848-8233 Telex 336 311
See You At NCC—Booth 31

NAME________ POSITION________
COMPANY________
ADDRESS________
CITY________ STATE/ZIP________
☐C200 ☐C201 ☐C202 ☐C207 ☐D302 ☐S1 ☐S2
☐S3 ☐S10B ☐SB1 ☐SB2 ☐SB3 ☐SB5 ☐SB6 ☐SB7
☐charge my ☐Visa ☐M.C. ☐Amer Express
Number________ Exp. date________
Signature________ ☐Send Catalogue

Circle 356 on inquiry card.

Listing 1: This BASIC program is the complete listing of the text editor.

```
10 REM EDITOR, VERSION 6, AS OF 1600 HOURS, 1/31/78
20 REM WRITTEN BY F.R. RUCKDESCHEL
30 REM 773 JOHN GLENN BLVD.
40 REM WEBSTER, NEW YORK 14580
50 REM NEW PARAGRAPH CHARACTER= &
60 PRINT "ENTER MAXIMUM LINE LENGTH",
70 INPUT M1
80 PRINT "ENTER LETTER LENGTH",
90 INPUT M2
100 DIM A$(M2),G$(M1 ),B$(2000),E(300)
110 PRINT "INPUT BEGINNING STORAGE LOCATION (DECIMAL): ",
120 INPUT Y9
130 REM TEST TO CHECK IF LETTER IS NEW
140 PRINT "NEW LETTER (N) OR RESTART (R): ",
150 INPUT G1$
160 IF G1$(1,1)="N" THEN GOTO 190
170 GOSUB 2010
180 GOTO 380
190 A$=""
200 G$=""
210 L=0
220 B$=""
230 PRINT "GET LETTER FROM FILE? (Y/N): ",
240 INPUT H$
250 IF H$(1,1)<>"Y" THEN GOTO 300
260 PRINT "INPUT NAME OF LETTER: ",
270 INPUT Z$
280 GOSUB 1530
290 GOTO 630
300 REM START OF NEW LETTER
310 L=L+1
320 PRINT %4I,L,
330 INPUT G$
340 IF G$="]" THEN GOTO 630
350 A$=A$+G$
360 GOTO 310
370 REM ********************
380 REM ENTER EDIT MODE
390 PRINT "EDIT SUBROUTINE"
400 PRINT "ENTER LINE NUMBER TO BE EDITTED",
410 INPUT L
420 REM L=0 IS ESCAPE EDIT
430 IF L=0 THEN GOTO 630
440 REM ADDITION AT FRONT OF LETTER?
450 IF L<2 THEN GOTO 1680
460 REM ADDITION TO END OF LETTER?
470 IF L>N-2 THEN GOTO 1790
480 Z=2*INT(L/2)
490 REM INSERTION? F CHANGE IS DEFAULT
500 IF L<>Z THEN GOTO 1900
510 PRINT %4I,L,A$(E(L-2)+1,E(L))
520 PRINT %3I,L,
530 G$=""
540 B$=""
550 INPUT G$
560 E(0)=1
570 IF G$(1,1)<>"]" THEN B$=B$+G$
580 IF G$(1,1)<>"]" THEN GOTO 550
590 IF E(L)+1>LEN(A$) THEN A$=A$+" "
600 A$=A$(1,E(L-2))+B$+A$(E(L)+1,LEN(A$))
610 GOTO 630
620 REM ********************
630 REM JUSTIFICATION ROUTINE
640 PRINT "COLUMN WIDTH",
650 INPUT W
660 PRINT "ENTER PAGE LENGTH: ",
670 INPUT P
680 C=0
690 S=1\N=2\H=0
700 PRINT "WANT LINES NUMBERED? (Y/N): ",
710 INPUT H1$
720 IF H1$(1,1)="N" THEN H=1
730 IF H=1 THEN J9=1
740 IF H<>1 THEN GOTO 780
750 FOR K=1 TO 7
760 PRINT
770 NEXT K\GOTO 870
780 PRINT "LINE NUMBER RANGE: (LOW, HIGH)",
790 INPUT N8,N9
800 J9=0
810 PRINT "WANT JUSTIFIED TEXT? (Y/N): ",
820 INPUT J9$
830 IF J9$="Y" THEN J9=1
840 FOR K=1 TO 5
850 PRINT
860 NEXT K
870 T=S
880 IF T>LEN(A$) THEN GOTO 1220
890 S=S+W
900 REM NEW PARAGRAPH FLAG
910 F=0
920 IF S>LEN(A$) THEN S=LEN(A$)
930 IF J9=0 THEN GOTO 1030
940 REM TEST FOR PARAGRAPH CHARACTER
```

key disadvantages to using this feature are that more memory is required and considerable time is consumed performing the extra storage function, which is relatively slow.

In the following sections the fundamental features of the editor program will be discussed in some detail. It will become apparent that the program does not have many frills, but still has considerable utility when one becomes practiced in its operation. A sufficient number of examples are given such that the user should not have to experiment much to determine how the program responds when kicked in a particular way. Following this are comments on software items which should aid in modifying the present program for use on other machines. The program as presented here suits my needs, but is probably deficient with respect to specific uses. However, the structure is sufficiently modular and sprinkled with comment statements such that the addition of new capabilities should be possible without a total revamping.

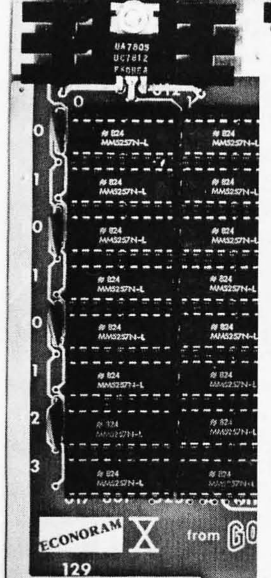
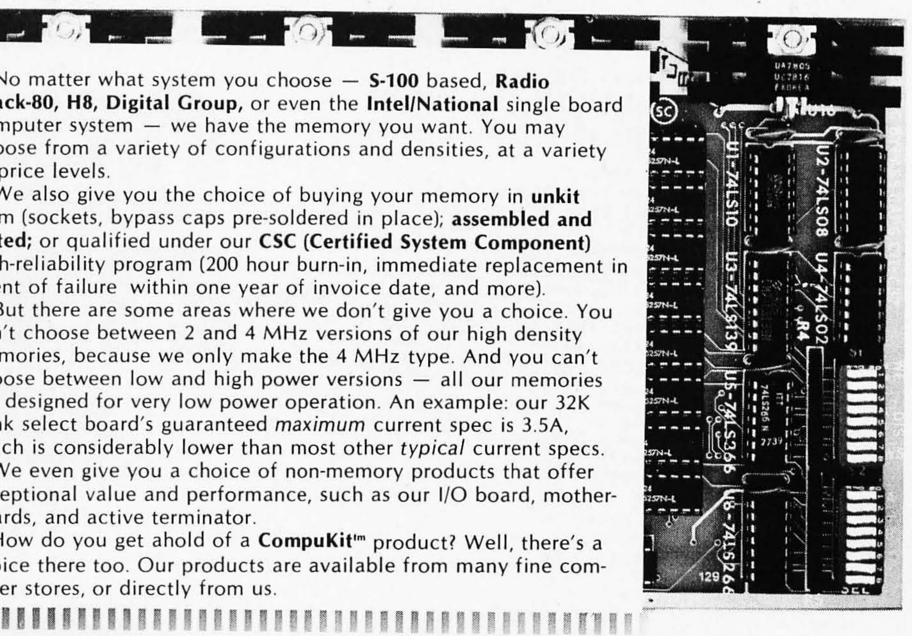
Editor Features

The object of the editor program is to generate a text oriented file which can be corrected and expanded with a reasonable level of ease. As mentioned earlier, the file may be stored on diskette, tape, or in active memory. The file can be recalled and printed out in two general formats, justified and unjustified. In the justification mode, by definition, lines are ended between words. The chosen line length may be any size as long as it is longer than the longest word in the text and shorter than the maximum line length allowed by the software. In the unjustified format, printing fills the entire line, regardless of whether or not a word is broken.

In either justification format the program will list selected line number groups using only even numbers. This allows the user to insert lines, when in the edit mode, by giving them odd numbers. Inserted lines can be any number of characters long within the constraints of memory or dimension. The program renumbers and reformats after each editing exercise.

Upon listing, it is the choice of the user to have the lines numbered or not. If the decision is not to number, the entire text will be printed. Otherwise a line number range can be chosen. In either case, the program goes through the entire text, reformats and assigns line numbers. In the justification mode defined above this may consume a significant length of time as tests for the gaps between words and new paragraph identifiers must be made. In the un-

for memory, the word is "CHOICE"

No matter what system you choose — **S-100** based, **Radio Shack-80, H8, Digital Group,** or even the **Intel/National** single board computer system — we have the memory you want. You may choose from a variety of configurations and densities, at a variety of price levels.

We also give you the choice of buying your memory in **unkit** form (sockets, bypass caps pre-soldered in place); **assembled and tested;** or qualified under our **CSC (Certified System Component)** high-reliability program (200 hour burn-in, immediate replacement in event of failure within one year of invoice date, and more).

But there are some areas where we don't give you a choice. You can't choose between 2 and 4 MHz versions of our high density memories, because we only make the 4 MHz type. And you can't choose between low and high power versions — all our memories are designed for very low power operation. An example: our 32K bank select board's guaranteed *maximum* current spec is 3.5A, which is considerably lower than most other *typical* current specs.

We even give you a choice of non-memory products that offer exceptional value and performance, such as our I/O board, motherboards, and active terminator.

How do you get ahold of a **CompuKit**™ product? Well, there's a choice there too. Our products are available from many fine computer stores, or directly from us.

Name	Storage	Buss	Design	Speed	Configuration	Unkit	Assm	CSC
Econoram II	8K X 8	S-100	static	2 MHz	2-4K	**$149**	**$164**	N/A
Econoram IV	16K X 8	S-100	static	4 MHz	1-16K	**$295**	**$329**	**$429**
Econoram VI	12K X 8	H8	static	2 MHz	1-8K, 1-4K	**$200**	**$270**	N/A
Econoram VII	24K X 8	S-100	static	4 MHz	2-4K, 2-8K	**$445**	**$485**	**$605**
Econoram IX	32K X 8	Dig Grp	static	4 MHz	2-4K, 1-8K, 1-16K	**$649**	N/A	N/A
Econoram X	32K X 8	S-100	static	4 MHz	2-8K, 1-16K	**$599**	**$649**	**$789**
Econoram XI	32K X 8	SBC	static	4 MHz	2-8K, 1-16K	N/A	N/A	**$1050**

BANK SELECT MEMORIES (for **Alpha Micro Systems, Marinchip,** etc.)

Econoram XII-16	16K X 8	S-100	static	4 MHz	2 ind. banks*	**$369**	**$419**	**$519**
Econoram XII-24	24K X 8	S-100	static	4 MHz	2 ind. banks*	**$479**	**$539**	**$649**
Econoram XIII	32K X 8	S-100	static	4 MHz	2 ind. banks**	**$629**	**$699**	**$849**

*addressable on 8K boundaries **addressable on 16K boundaries

Econoram is a trade mark of **Godbout Electronics.**

ALSO AVAILABLE: 16K MEMORY EXPANSION CHIP SET $109 (3/$320)

For **Radio Shack-80, Apple, Sorcerer** machines. 250 ns chips for 4 MHz operation, DIP shunts, 1 year limited warranty. With easy-to-follow instructions.

ACTIVE TERMINATOR KIT $34.50

Our much imitated design plugs into any S-100 motherboard to reduce noise, ringing, overshoot, and other buss-related problems.

DUAL CHANNEL/DUAL FUNCTION S-100 I/O BOARD

$189 unkit, $249 assm

This one does things the others can only dream about. Two independently addressable serial ports with full RS232C current loop (20 mA) and TTL signals on both ports, crystal controlled Baud rates to 19.2 KBaud, hardware UARTS, compatible with 2 or 4 MHz systems, low power consumption, and a whole lot more . . . we'll be glad to give you more info, just write.

2102 MEMORY SPECIAL 10/$9.90

That's only **99c** each for *very* low power, static 1K RAMs suitable for use with any 2 MHz system. Price only good on orders of 10 or more.
16 pin lo-profile sockets also available at the special price of **12/$2.00.**

1791 MOS LSI DUAL DENSITY DISC CONTROLLER CHIP $59

Just in. Prime Western Digital parts, with pinout and data.

HEATH H8 BARE BOARD MEMORY SPECIAL — $35!

Don't need the full 12K of our standard H8 memory? We now offer the board, mounting bracket, edge connector, and schematic for only **$35.** Populate it with a few support chips and readily available, low cost 2102s to build your memory up to a full 12K whenever your budget permits.

11 SLOT MOTHERBOARD UNKIT

With 11 edge connectors pre-soldered in place for easy assembly, as well as integral active terminator circuitry. Dimensions: 8.5" X 11" **$90**

18 SLOT MOTHERBOARD UNKIT

Same as above, but 18 slot version with 18 edge connectors. Dimensions: 8.5" X 16.7" **$124**

TERMS: Allow 5% shipping, excess refunded. Cal res add tax. VISA® /Mastercharge® call our 24 hour order desk at **(415) 562-0636.** COD OK with street address for UPS (UPS COD charge applies). Prices good through cover month of magazine. Orders under $15 add $1 handling.



BOX 2355, OAKLAND AIRPORT, CA 94614

from



FREE FLYER: We'll be glad to tell you more than the space of this ad allows. Just send your name and address, we'll take care of the rest. If you're in a hurry, enclose 41c in stamps and we'll send it out first class.

justified mode the operation is very fast,

In addition, changing the first character in the text is a little cumbersome; it is suggested that the very first character be a blank to mitigate this problem.

When any of the above failures occur there is a good chance that recently edited work may be lost. For example, when a program fails, most BASICs enter the direct command mode. If the program is rerun, most likely the string variables will be cleared. The temporary save and retrieval subroutines were included to reduce the impact of such errors. If the error occurs during the edit mode, the last form of the text (before the edit mode was entered) may be retrieved by rerunning and answering the initialization appropriately (restart: R). If the failure occurs during a disk or a tape load, unless a large crash occurs, the most recent text version may be recovered upon restart.

Additional Notes

There are several statements employed in North Star BASIC which must be modified if the editor program is to be used with another interpreter. The following is a list of the types of changes required if this program is to be translated into a BASIC similar to that written by Microsoft and distributed by MITS:

FILL<>,<> same as POKE<>,<>
EXAM(<>) same as PEEK(<>)

% Denotes a printing format (for example, nFm which is similar to the FORTRAN real format nFm). This can be simulated using the INT function along with some multiplication and division. In the more advanced versions of BASIC there is usually an equivalent format statement.

, Denotes a continued print statement without a carriage return and line feed. In MITS BASIC one would use a semicolon.

A$(M,N) With respect to the editing functions as implemented by the program, this is the most important difference between the North Star and Microsoft interpreters. For the task at hand the North Star form is preferred because of ease of use. In North Star BASIC A$(M,N) represents a substring of A$ which runs from character position M to character position N, thus having length N–M+1. To accomplish this in Microsoft BASIC one would use MID$(A$,M, N–M+1). Although a little more cumbersome, the Microsoft string function would suffice if it were not for the important variance that North Star BASIC allows arbitrarily long string lengths whereas Microsoft limits the string length to 255 characters. However, this limitation may be overcome by string subscripts. Note that string dimensions and subscripts are permitted in Microsoft BASIC, but not in North Star. Having to resort to subscripts is certainly an inconvenience and is a small pain to program. North Star BASIC certainly has an advantage in that respect.

The most difficult editing function to perform is the alteration of a text segment as the set of characters to be changed may extend over more than one text line. Generally it is best to attack the last line of the modification first. The corrections should then proceed toward the lower line number. The reason for doing this is that the text is reassembled upon each editing pass. If a change is made in a given line, all the lines above that point are altered, whereas all the lines below that point maintain their structure if the same line width format is chosen.

Inserting text is usually easy. As a redundant word of caution, remember to place a blank in the last character position in the file. Also, as there is some awkwardness in changing the very first character in the file, having a blank there is also helpful.

Conclusion

The limited capability editor program presented in the previous sections has the advantages of being easy to use as well as being somewhat portable. It was really meant to operate with the North Star Disk System, but can be translated for use on other BASIC oriented machines. Its utility is exemplified by the present article which served as a test case. In this application the article was broken up into files of 12 blocks (12 by 256 characters) or less to ease the strain on active memory. Thus, in effect, there is no limit to the document length which can be handled. Incidentally, 12 blocks corresponds to approximately half a typeset magazine page.

The true power of this program is apparent when large portions of the text creation and editing are done on a video display system (it is easy to get into the swing of rapidly typing additions and changes). However, occasional hard copy listings are invaluable for leisurely reviewing. A near ideal combination would seem to be a video display for editing and a fast hard copy terminal for printing. The denser and faster the video display the better.■

Copies of the software shown on listing 1 are available from the author on diskettes in North Star format for $9.95 each (New York State residents add sales tax).

Bubble Memories

A Short Tutorial

A I Halsema
32014 Grenville Ct
Westlake CA 91361

In the first quarter of 1977 Texas Instruments announced the availability of a 92,000 bit magnetic bubble memory, making bubble memories a commercial reality ten years after the effect was discovered at Bell Laboratories. The bubble memory provides nonvolatile, medium speed data storage at a price close to that of the floppy disk, but without either moving parts or the problems of reliability that moving parts entail. Future memories will provide storage densities of up to one million bits per chip and faster data rates, as the technology matures.

Bubble memories are shift registers that move magnetic domains representing binary data using rotating magnetic fields. The bubble memory integrated circuit is made of a magnetically reactive material such as garnet and has implanted in it a bubble generator for writing data, bubble detectors for reading data, bubble annihilators for erasing data and replicators to provide nondestructive readout. Bubbles move in loops made of small bars of permalloy. As the magnetic field driving circuit rotates, the permalloy bars change their magnetic bias, attracting or repelling the bubbles, as shown in figure 1. This creates a shift register. The Texas Instruments TBM 0101 memory contains 157 minor loops of 641 bubble positions for data storage, and a major loop of 640 bubble positions for reading and writing data. This configuration is shown in figure 2.

In operation, data is written into the major loop at the bubble generator. The bits so written are shifted along the major loop until they are adjacent to the minor loops where they are to be stored. The bubble transfer line is then activated and the bubbles are shunted onto the minor loops. Of the 157 minor loops on the device, 144 are guaranteed to be good, so the user must avoid writing into any of the 13 minor loops declared bad at the factory. Each memory device will be shipped with a map of the bad loops written into the device at the factory. This map should be written into programmable read only memory to control access of the minor loops. The controller integrated circuit offered by Texas Instruments will have an input to cause it to ignore bubbles and may be driven by the read only memory. To read data, the minor loops are rotated until the desired 144 bubbles are adjacent to the major loop. The transfer line is activated and the bubbles are shunted out of the minor loops and onto the major loop. The bubbles are then shifted along the major loop until they reach the combination replicator and annihilator.

If a replicate pulse is applied to the circuit, the bubble is duplicated. While one copy goes on to the detectors and eventual

Update on Bubble Memories

In the third quarter of 1978, after this article was written, Texas Instruments announced a new magnetic bubble memory which is capable of storing 250,000 bits of information. The new circuit has 252 minor loops consisting of 1137 bubble positions of which 224 are guaranteed. The memory has an access time of 7.3 ms for the first bit of the 224 bit page.

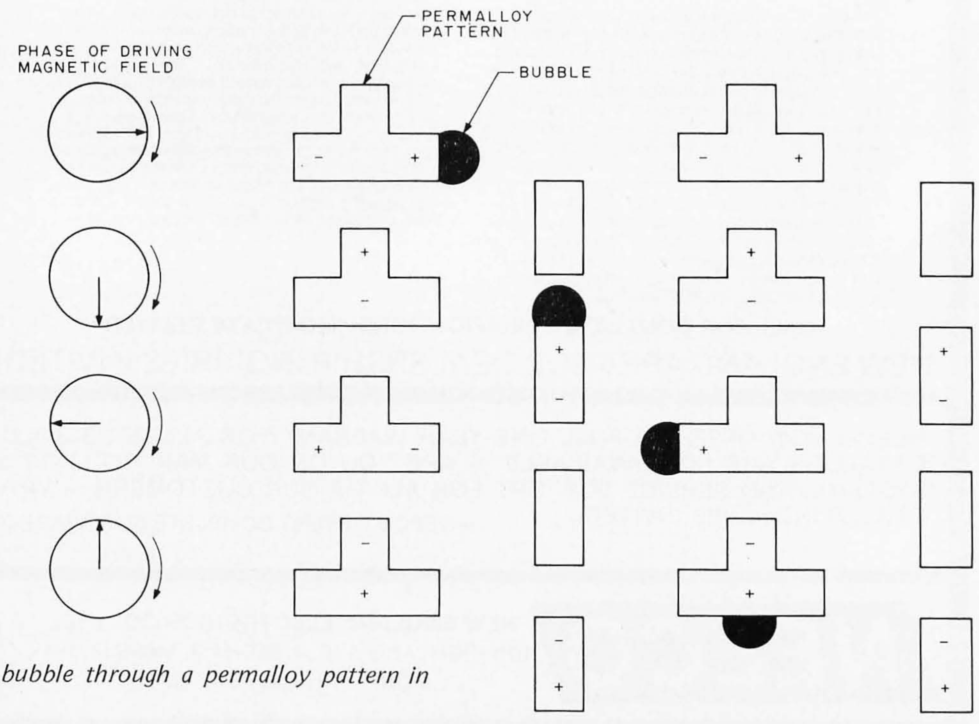


Figure 1: Movement of a magnetic bubble through a permalloy pattern in response to a rotating magnetic field.

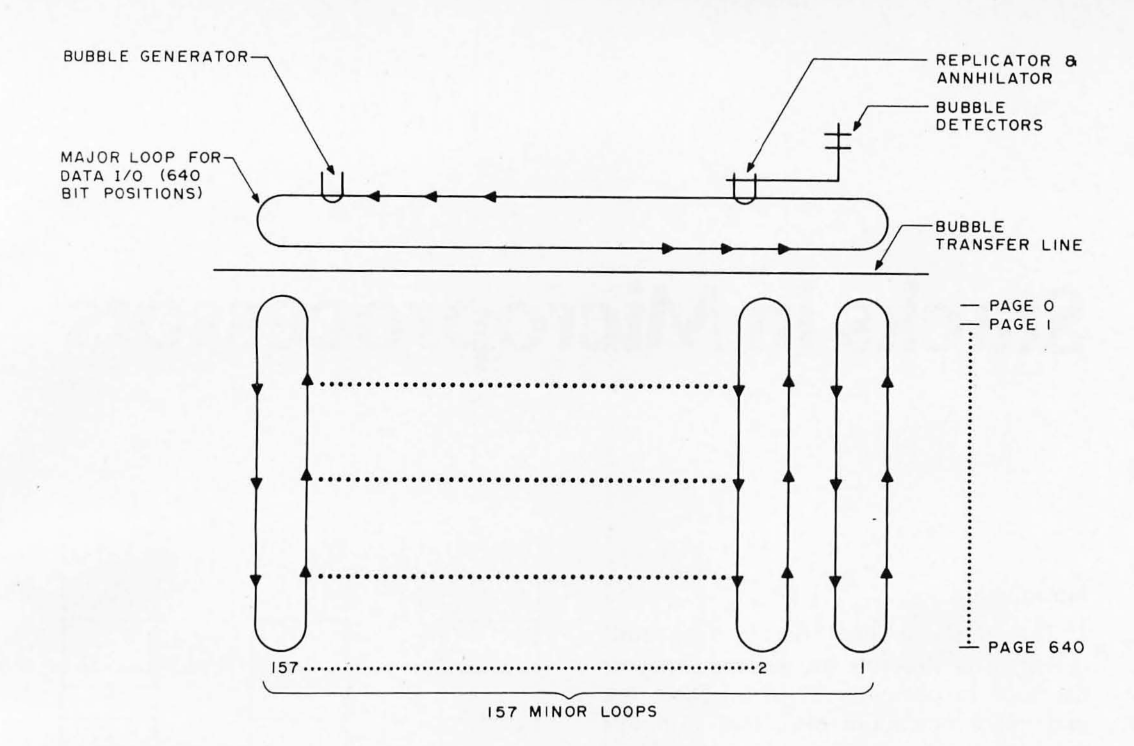


Figure 2: Logical layout of the Texas Instruments TBM 0101 bubble memory. The 157 minor loops are used for data storage. Each loop contains 641 bit positions.

destruction, the other bubble continues around the major loop until it is shunted back onto its minor loop, thus providing nondestructive readout. The bubble that is passed on to the detectors activates dual Hall effect devices that vary their electrical resistance with varying magnetic fields. Two of these detectors are supplied in order to cancel noise through the use of a detector bridge connected to a differential amplifier. The detectors have a nominal resistance of 1100 ohms and are matched to within ±10 ohms. A single bubble passing under the two detectors produces a 7 mV signal with two positive and two negative peaks. Care must be taken when laying out circuit boards to avoid long detector signal lines and cross-coupling of control signal noise.

The rotating magnetic field for shifting the bubbles is produced by driving the coils built into the circuit with signals that are 90° out of phase at a maximum frequency of 100 kHz. This technique is shown in figure 3. For standby low power operation, the coil drive may be turned off with no loss of data.

Data in the memory is organized in the form of 18 byte pages and 641 pages per integrated circuit for a total of 11,538 bytes. Using the TMS 9916 controller, any byte within a page may be randomly accessed, although the pages are serially accessed. Each minor loop contributes a bit to the formation of a page, so that the 144 bits of a page are distributed over 144 minor loops.

Prior to removing power from the memory system, the user must be sure that page zero is rotated back into the page zero position. If this is not done, the next time power is applied it may not be possible to locate page zero for proper addressing. This is because there are no physical indices to mark the beginning of the medium. If the TMS 9916 controller is used, power failure detection circuitry will initiate data positioning automatically. This process requires 12.8 ms, so the system's power supplies should remain in regulation for at least this long.

Considering the high price of the magnetic bubble medium, it is unlikely that it will be used in a removable form as is the case with the floppy disk. Instead, they will probably be built into a cabinet (or even a mainframe) and used as a drum would be. An 11 K byte drum may seem small, but remember that single circuits with 3 to 10 times the capacity of the TBM 0101 are on their way.■

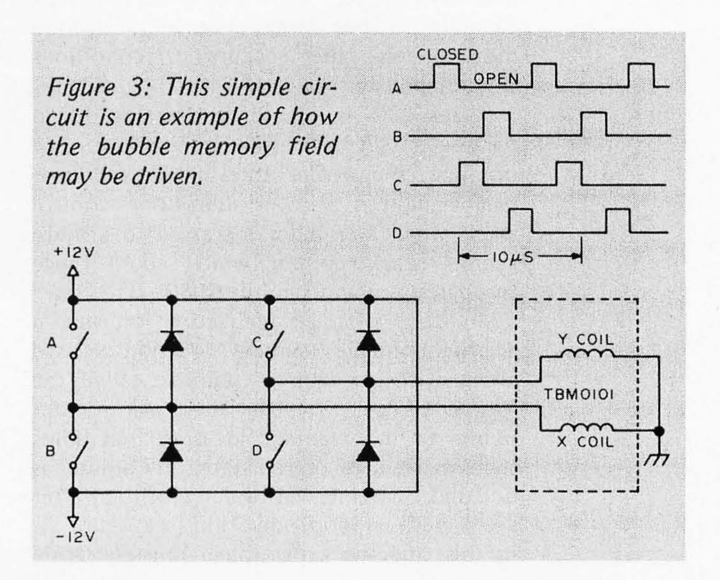


Figure 3: This simple circuit is an example of how the bubble memory field may be driven.

Stacks in Microprocessors

T Radhakrishnan
Dept of Computer Science
Concordia University
Montreal CANADA

M V Bhat
Pratt & Whitney Aircraft, Canada
Engineering Computing
Longuenil Quebec CANADA

Introduction

The stack or the "last in first out" (LIFO) data structure has become an essential tool in computer systems. There are two major operations associated with this data structure:

PUSH: places a new data item on top of the existing ones in the stack.

POP: removes the topmost element of the stack for succeeding operations.

A spring loaded plate holder in a cafeteria is a good example of a "stack," since addition and removal of items occur at the same end in a last in first out sequence (see figure 1).

When the capacity of a stack is "n" items, then n+1 consecutive PUSH operations will cause the stack to overflow. Similarly, popping an empty stack creates an underflow. Even though stack underflow may not occur intentionally, programmers should account for this condition. Stack overflow is more probable when the stack capacity is not large enough to accommodate all the occurring conditions simultaneously.

Stack size is one of the major design parameters in processor architecture. For instance, the earlier Intel 8008 processor had a built-in 7 level subroutine control stack which was later increased to a more general stack pointer which could range throughout memory in the 8080.

In the software realization of stacks, a programmable memory location is used along with an address pointer, called the "stack pointer" or SP. The stack pointer points to the memory location that holds the top element of the stack; the pointer is updated (incremented or decremented) after every push or pop operation (see figure 2). In this case the programmer must set aside a portion of the main memory to accommodate the stack. Consequently, the stack capacity is determined by the free space in the main memory and is more flexible. In figure 2 the occupied portion of the stack grows from low to high memory addresses. Hence, the PUSH operation increments the stack pointer and the POP operation decrements it. It is not difficult to introduce the stack overflow and underflow conditions in the above simulation.

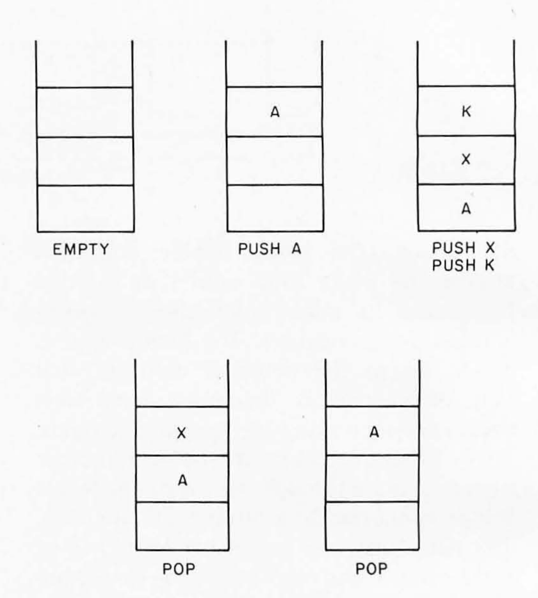


Figure 1: A sample 3 word stack. A PUSH command causes one piece of data to be "PUSHed" onto the stack; the resident data is pushed downward to make room. Similarly, a POP command removes the topmost piece of data and shifts the rest of the stack upward.

In another realization of stacks, a set of n registers constitutes a stack. Every POP operation takes the data item from the topmost register; the data in each stack location is then shifted upward. The PUSH operation shifts the stack contents down one place

ONE PACKAGE DOES IT ALL

Includes these Application Programs . . .

Sales Activity, Inventory, Payables, Receivables, Check/Expense Register, Library Functions, Mailing Labels, Appointments, Client/Patient Records

RANDOM, MULTI-KEY RECORD RETRIEVAL under CP/M, CDOS, IMDOS, ADOS . . .

SELECTOR III ALLOWS INSTANT RECALL OF ANY RECORD USING ANY INFORMATION ITEM IN THE RECORD. That statement deserves re-reading, because that ability makes SELECTOR III the most powerful Date Base Management System in microcomputers today!

With **SELECTOR III** you can...

- define a record format, assign retrieval keys, and begin entering data in minutes.
- create sorted pointers to records matching your specif or range of requirements.
- automatically generate reports with control-break summaries and unlimited variety.
- bring an application online in hours instead of months.

SELECTOR III comes complete with eight application programs that perform the tasks listed at top of page. And, since it's distributed in source code form, you can easily add subroutines to do specific computations or file updates.

SELECTOR III runs under CBASIC Vers. 1 or 2, and is priced at $295. SELECTOR III-C2 is dedicated to Vers. 2 only, runs about twice as fast, and costs $345.

Both systems are available in a variety of CP/M, diskette size and density formats including IBM 8"; North Star; Micropolis; TRS-80; Processor Tech Helios II; Altair; iCOM; Dynabyte; Imsai; and others.

Available from computer stores nationwide:

LIFEBOAT Associates
2248 Broadway, Suite 34,
New York, N.Y. 10024 • (212) 580-0082

Or order direct from

MICRO-AP
9807 Davona Drive, San Ramon, CA 94583
(415) 828-6697

Circle 213 on inquiry card.

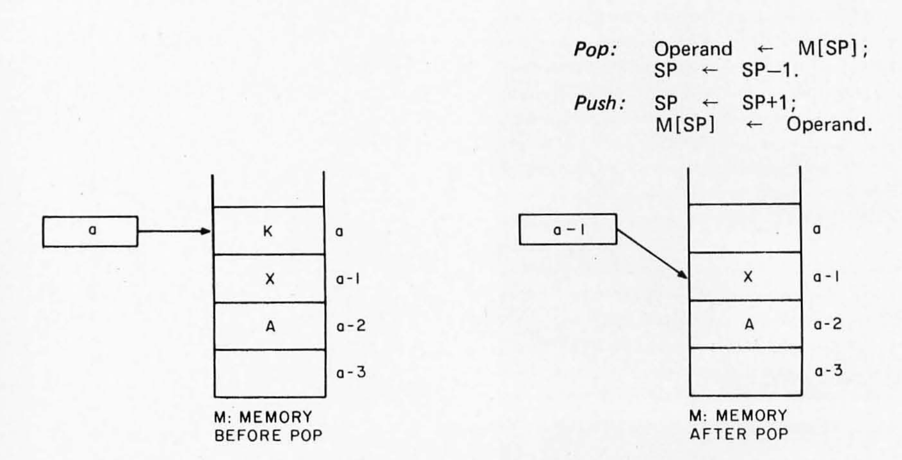


Figure 2: A software simulation of the pushdown stack. Operation of the stack is identical to the hardware stack (see figure 1), except that there is no dedicated hardware involved. Instead, a program creates a stack pointer in memory which points to the current location of the top of the stack.

Processor	Hardware Stack or Stack Pointer	Stack Oriented Instructions	Remarks
1. 8080	16 bit stack pointer	a) Push register pair into stack b) Pop register pair from stack c) Push/Pop processor status word d) Exchange stack top with register pair (H,L) e) Load SP from register pair (H,L)	
2. Z-80	16 bit stack pointer	a) All the instructions of Intel 8080 b) Push/Pop the (two) index registers	
3. M6800	16 bit stack pointer	a) Push/Pop the (A or B) accumulator b) Load SP from memory c) Store SP into memory d) Transfer index register contents to SP e) Transfer SP into index register f) Increment/Decrement SP	
4. RCA 1802	16 bit stack pointer	a) Increment/Decrement the selected register (SP) b) Push/Pop the working (D) register c) Load the D register into left or right half of SP	Any of the 16 registers can be used as a SP
5. PACE	Hardware stack 8 16 bit words	a) Push/Pop program counter b) Push/Pop the specified register c) Exchange the contents of the register with SP d) Push/Pop the flag register	Stack overflow Underflow Interrupts are provided
6. IMP-8C	Hardware stack 16 8 bit words	a) Push/Pop the selected accumulator into stack b) Exchange the stack top with the selected accumulator c) Push/Pop the status flags into the stack	No overflow Underflow Interrupts

Table 1: Stack features of some common microprocessors. The stack is a storage place in a computer designed to hold pieces of data in serial order. "PUSHing" an element onto the stack causes the existing elements in the stack to be moved downward, in much the same manner as a spring loaded plate holder found in restaurants. "POPing" an element from the stack removes the most recent addition to the stack for use. Because of these two features, the stack operation is often referred to as "last in first out," or "LIFO."

and adds the new data item. In this approach, reading from and writing to the data structure occur only with the topmost register. Inter-register transfers can be achieved in parallel during the same clock period. The stack facility available with IMP-8C microprocessor, an example of this type, has a capacity of 16 words. This method of realization is known as the *fixed top* (figure 1) in contrast to the *moving top* approach explained earlier (figure 2). The flexibility associated with the latter can be combined with the speed advantage of the former as is done with PACE microprocessors (see table 1).

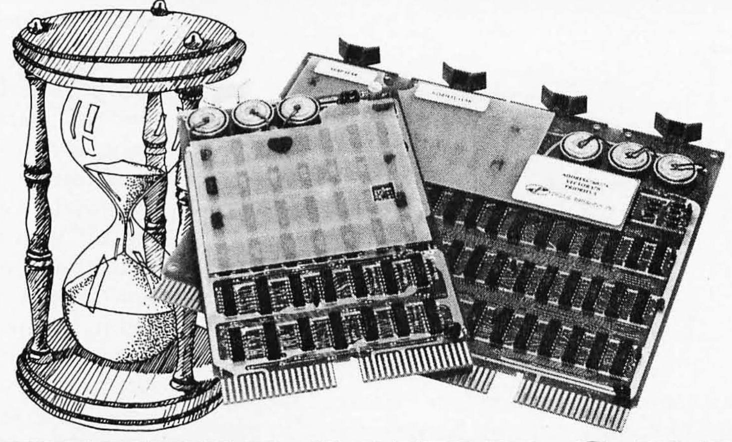
Most modern processors provide one or more registers to hold stack pointers. For example, there is one stack pointer register in the Intel 8080 and there can be as many as 16 stack pointers in the RCA 1802 processor (see table 1). The pop and push instructions update the SP registers automatically. The architecture and the stack oriented instructions differ widely among the various processors, and table 1 gives details of some of the common ones.

Typical Applications of Stacks

Suppose a routine A calls another routine B at some point a in A. Similarly, let B call C at point b. The addresses a+1 and b+1 are the return addresses where execution control will return from the called routine. It is evident from figure 3 that the return addresses are used in the reverse order of their sequence of occurrence. The labels c1, c2, c3 in figure 3 stand for the first, second and third calling of routines, and r1, r2, r3 stand for the first, second and third returns from the called routines. This last in first out (LIFO) nature of the use of return addresses in multilevel calling is commonly implemented with stacks. Simple extensions have been devised to pass the parameters along with these return addresses using the stack structure (see reference 1).

The calls shown in figure 3 could also be considered as calls to service routines due to asynchronous interrupt signals. In the latter case, the return addresses are not predetermined address points, but are instead the contents of the program counter. However, the last in first out nature of the return addresses remains valid. The call due to an interrupt creates a new process, and hence the status of the current process (process status word, flags, etc) has to be additionally

TIME after TIME



BATTERY SUPPORTED CALENDAR CLOCKS

PDP-11*

TCU-100 • $495

- Provides month, day, hour, minute and second.
- Can interrupt on date/time, or periodic intervals.

TCU-150 • $430

- Provides year, month, day, hour, minute and second.
- Automatic leap year.
- Patches for RSX-11M, RT-11 FB/SJ VO2, VO3 and UNIX.

LSI-11/2*

TCU-50D • $295

- Provides month, day, hour, minute and second.
- Dual size board.
- Patches for RT-11 SJ/FB VO2, VO3B.

Lockheed SUE

TCU-200 • $550

- Provides year, month, day, hour, minute, second and milli-second.
- Interval interrupts between 1/1024 seconds and 64 seconds.

Computer Automation (Naked Mini)

TCU-310 • $385

- Provides year, month, day, hour, minute and second.



Multi-Bus**

TCU-410 • $325

- Provides year, month, day, hour, minute and second.
- SBC/BLC compatible.

HP 2100

TCU-2100 • $395

- Correct time restored after power failure.
- Compatible with the HP TBG card.

Serial Clock (RS 232 or 20 mA)

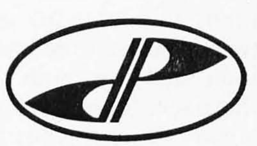
SLC-1 • $575

- Connects between any terminal and host computer.
- Provides date, time and more!

All Digital Pathways TCUs have on board NICAD batteries to maintain time and date during power down. Timing is provided by a crystal controlled oscillator. Prices are U.S. domestic single piece. Quantity discounts available.

For more information on these products, contact:
Digital Pathways Inc.
4151 Middlefield Road
Palo Alto, CA 94306
Phone: (415) 493-5544

**Trademark of Digital Equipment Corporation*



***Trademark of Intel Corporation*

DIGITAL PATHWAYS

Circle 86 on inquiry card.

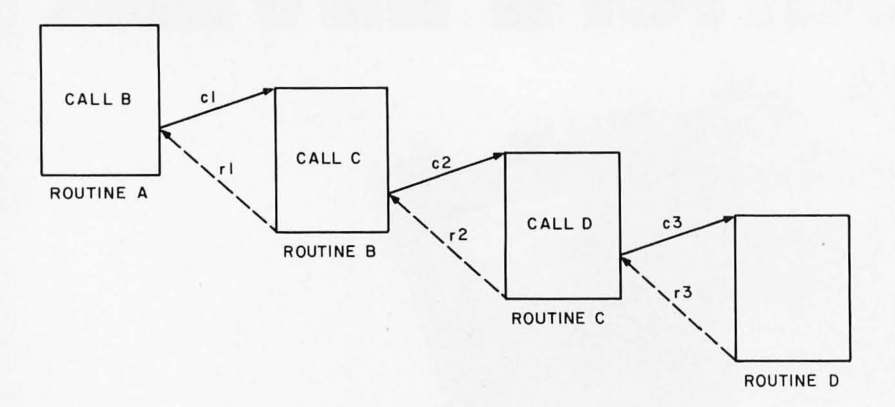


Figure 3: Diagrammatic representation of multilevel, or "nested," subroutines. The return address of each subroutine call must be remembered so that the program can return to the right place after the subroutine is completed. The "last in first out" nature of nested subroutines is such that the stack is a logical way to keep track of the return addresses.

saved. Some processors, like the IMP-8C, have instructions to push and pop status flags onto stacks. In other processors, this is done automatically when an interrupt occurs. Stacks in microprocessors, starting from the early Intel designs, have traditionally been used primarily for subroutine control and interrupt handling.

Another use of stacks, though one not much used in the hardware of processors, is in the compiling arithmetic expressions. Consider the following arithmetic expression:

A+BXC–D/E

In this form, the "operator" is between the two operands. This is known as *infix* notation. The form in which the operator follows the operands is called *postfix* or *reverse Polish* after the Polish logician J Luckasiewicz, who investigated the properties of this notation. The postfix equivalent of the above expression, which does not require any parentheses, is as follows:

AB+CXDE/–

Algorithms exist which use the stacks to convert arithmetic expressions from infix to postfix notation (see reference 2). Figure 4 shows a sample code for the above postfix expression; it is meant for a computer with stacks, and is used to evaluate arithmetic expressions. Operations such as ADD and SUB take the top two elements of the stack, perform the operation, and then push the result back onto the stack. Such a system is called a *stack computer*. Using this postfix notation, it is not hard to generate code for machines with single accumulators or for machines with multiple registers.

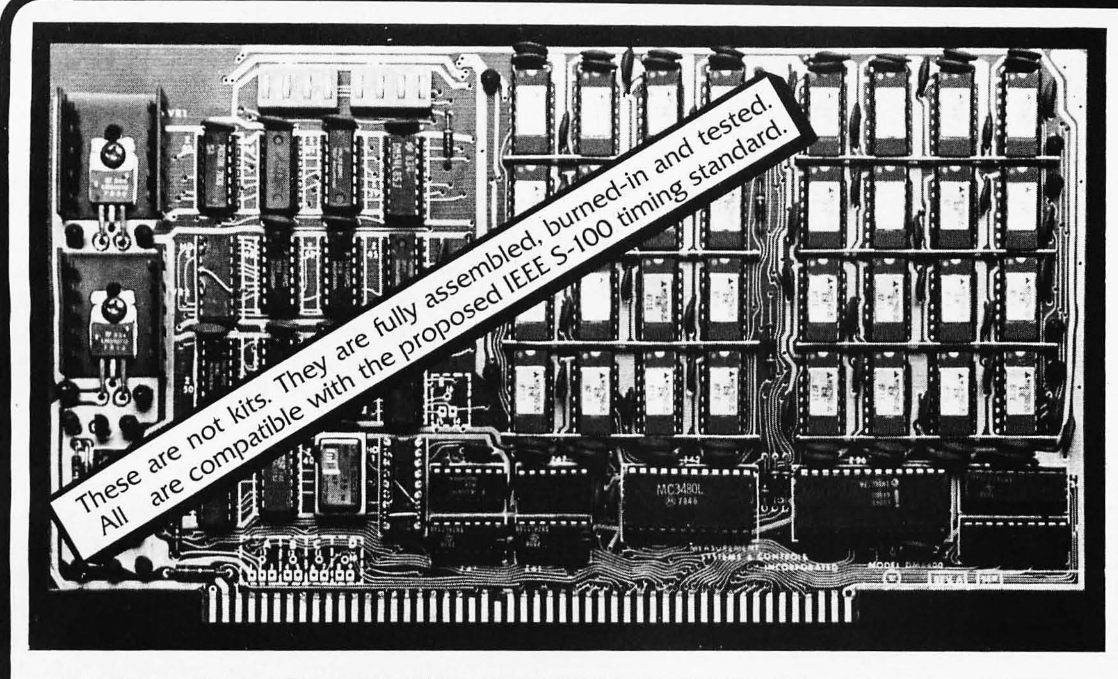
Stack Machines

Among the architectures with two stacks, two broad categories are evident. The first kind of machine provides stack features along with conventional architecture. This stack feature might be implemented through a hardware realized stack, a stack pointer register with a set of associated hardware instructions, or a complete software simulation using a memory location as the stack and its pointer. Some combinations of these three approaches are also present in some recent processor architectures. Most processors have some sort of stack facility and instructions to manipulate data with stacks or stack pointers.

The second kind of machine with stack facility can be called a *stack machine*. Its architecture is completely centered on stacks. The Burroughs B5500 and B6700, HP3000 and ICL2900 are examples of this category. In these machines, the three basic functions of process management, memory management, and data management of jobs are all stack oriented. Most of these archi-

Op Code	**Contents of Stack (read left to right)**
PUSH A	A
PUSH B	B,A
ADD	(A+B)
PUSH C	C,(A+B)
MPY	(A+B)*C
PUSH D	D,(A+B)*C
PUSH E	E,D,(A+B)*C
DIV	(D/E),(A+B)*C
SUB	(A+B)*C–(D/E)

Figure 4: Op code designed for use with Polish postfix notation on stack oriented computers. Polish notation is a method for rewriting expressions unambiguously by systematically segregating operators and operands. For instance, the expression used in this example appears as (A+B)XC–D/E in normal, or "infix" notation; the Polish postfix equivalent is AB+CXDE/. The latter can be directly used by a stack oriented computer, which automatically performs stack operations. (For example, a stack ADD instruction takes the top two elements of the stack, adds them together, and pushes them back onto the stack. The MULT, DIV and SUB operators work in the same manner.) The algorithm for evaluating the expression then reduces to examining each element in the Polish notation string from left to right, pushing it onto the stack if it is an operand and performing the operation if it is an operator.



• NEW PRODUCT •
64K bytes of fast, low power dynamic RAM. MEMORY BANK SELECT
$895.00
for model DMB 6400

64K bytes of fast, low power dynamic RAM.
$795.00
for model DM–6400

• NEW PRODUCT •
32K bytes of faster Static RAM which MEMORY BANK SELECT
$695.00
for model SMB 3200

Both our UNCommon Dynamic and our UNCommon Static RAMS have the following features and specifications:

- They are all GUARANTEED to be compatible with the following S-100 systems: CROMEMCO, IMSAI, ITHACA AUDIO, MITS, NORTH STAR, PROCESSOR TECHNOLOGY, TDL, TEI, VECTOR GRAPHICS, and other S-100 systems.

Both of Our UNCommon Dynamic RAM Series, the DMB-6400 and the DM-6400, feature:

- 64K bytes of dynamic RAM with on board transparent refresh.
- S-100 interface compatible, with crystal controlled timing INDEPENDENT of bus or processor timing.
- No wait states with 8080 or Z-80 to 4MHz. Up to 5MHz with 1 wait state.
- Memory selectable or deselectable in 4K byte increments.
- Low power, 8 watts maximum, in 64K byte configuration.

Our UNCommon Static RAM Series, the SMB-3200, features:

- Memory Bank select capabilities: Either two (2) 16K byte banks of memory, or one (1) 32K byte bank per board.
- 32K bytes of low power static RAM.
- No wait states with 8080, 8085, or Z-80 processors up to 5MHz.
- Addressable in 4K byte increments at 4K boundaries. Deselectable in 4K byte increments.

Announcing the Model DMB-6400 Series of UNCommon Dynamic RAMS. This Series incorporates the features which are standard in the DM-6400 Series and adds the following capabilities:

- MEMORY BANK SELECT which is compatible with CROMEMCO, NORTH STAR, and other systems using output port bank select.
- 256 ports selectable (including CROMEMCO 40H and NORTH STAR C0H)
- Eight (8) 64K bytes banks of memory per output port.
- Each 16K bytes of addressable memory space may be individually set to start at 0000, 4000, 8000, or C000 and can be set for any of the 8 banks on one selected output port. The bank memory size can be incremented from 16K bytes to 64K bytes in 16K increments, allowing 512K byte bank sizes.
- Total memory addressing to over 100 Megabytes by using different I/O ports for memory control.

All of our UNCommon Dynamic and Static Series RAMS feature the following:

- Inputs are RC filtered and buffered with 1 LS TTL load. Compatible with terminated or unterminated busses.
- Outputs are all tri-state.
- Disc compatible. DMA compatible.
- Phantom memory selectable on pin 67.
- DIP switch selectable addressing.
- Reliability — all boards fully tested and burned-in.
- FULL DOCUMENTATION — Schematics, layout, parts list, theory of operation, timing diagrams, and option selection.
- Industrial quality design, components and construction. Glass epoxy boards. Fused, double solder masks. Silk screened legends. Gold plated connector contacts. All IC's socket mounted.
- Guaranteed performance on parts and labor for one year.

● UNCommon Dynamic RAMS with MEMORY BANK SELECT.
- ☐ DMB 6400/64K RAM — $895.00
- ☐ DMB 4800/48K RAM — $795.00
- ☐ DMB 3200/32K RAM — $695.00

● UNCommon Static RAMS with MEMORY BANK SELECT.
- ☐ SMB 3200/32K RAM — $695.00

● UNCommon Dynamic RAMS
- ☐ DM 6400/64K RAM — $795.00
- ☐ DM 4800/48K RAM — $695.00
- ☐ DM 3200/32K RAM — $595.00

Delivery: stock to 30 days.
• PRICES SUBJECT TO CHANGE WITHOUT NOTICE. •

NEW PRODUCTS
*Available soon.

MEASUREMENT
systems & controls
incorporated

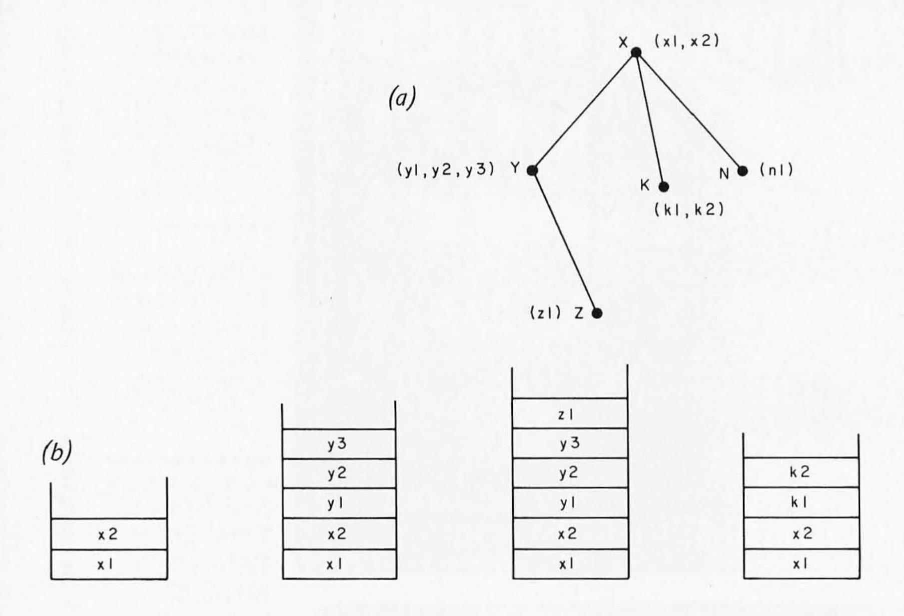
867 North Main Street
Orange, California 92668
Telephone: (714) 633-4460

For the above products see your local dealer or order directly.

ORDERS — BA, VISA, MASTER CHARGE, money orders, or personal checks. Please allow 14 days for checks to clear bank. California residents please add 6% sales tax to your order.

All orders shipped postpaid. All orders in U.S. funds. Please add 10% on all orders outside the U.S.A. and Canada.

Circle 215 on inquiry card.



(c)

```
X: Begin
   Integer x1, x2;
      .
      .
      .
    Y: Begin
       Integer y1, y2, y3;
          .
          .
          .
       Z: Begin
          Integer z1;
             .
             .
             .
          End;        (Z)
          .
          .
       End;    (Y)
   .
   .
   K: Begin
      Integer k1, k2;
         .
         .
      End;     (K)
   .
   .
   .
   .
   N: Begin
      Integer n1;
         .
         .
         .
      End;     (N)
   .
End;    (X)
```

Figure 5: A block structured program. Programs written in block structured languages can be visualized as tree structures (figure 5a). ALGOL and PL/I are examples of this type of language. The tree in this illustration shows how the program is structured. Figure 5b shows how the stacks in a stack oriented machine would look at various points of the program. Figure 5c shows the block layout of the program.

tectures support block structured languages similar to ALGOL or PL/I. A program written in a block structured language can be visualized as a tree structure; execution of the program traces some paths in this tree structure. The relationship between tree structures and stack data structures is well known (see reference 4). An example is shown in figure 5 along with "snapshots" of stacks holding the program variables. Because of the limited access points with stacks, certain extensions are required in stack machines to implement the array data structures. These extensions are of a different kind, such as the use of index registers for addressing. Similarly, to facilitate process and memory management, special software tools are used.

Computer systems and architectures can be appraised from three points of view: the languages available to users (application and system programmers), the operating system, and the hardware. These three areas are highly interrelated, and it is difficult to separate their capabilities. A few stack machine architectures are commercially available with facilities for multiprogramming and timesharing. The architecture of the Burroughs systems is such that the system software can be effectively written in a high level language. Stack machines have good and bad points. Their advantages are noticeable in block structured programming, which is becoming popular. As Doran points out (see reference 1), stack machines have proven to be successful. The increasing cost of software and the flexibility available through microprogramming indicates a trend towards stack machines or, at least, toward a greater use of stack features in computer architectures.

Conclusions

Developments in software and programming techniques during the past decade have proven the advantages of stack data structures. Microprocessors of recent origin provide adequate facilities to support this data structure. The provision of stack pointers is a compromise between the expensive and inflexible hardware stacks at one end and the inexpensive and flexible software simulation at the other end. Most microprocessors have stack pointers and a set of associated machine instructions.

Stack machines have certain advantages in higher level block structured programming and the implementation of operating systems. At present, programming with microprocessors is done mostly in machine or assembly language level. Large in-house software systems for microprocessors are not yet a reality. As a result, stack machine architectures are still in the realm of large machines.■

Acknowledgement

We gratefully acknowledge the help of K Venkatesh, research assistant, Computer Science Department of Concordia University, in the preparation of this manuscript.

REFERENCES

1. Doran, R W, "Architecture of Stack Machines" in *High Level Language Computer Architecture*, edited by Y Chu, Academic Press 1975.
2. Gries, D, *Compiler Construction for Digital Computers*, John Wiley & Sons, NY 1971.
3. McKeeman, W, "Stack Computers" in *Introduction to Computer Architecture*, edited by H S Stone, SRA Inc 1975.
4. Knuth, D E, *The Art of Programming, vol 1, Fundamental Algorithms*, Addison Wesley, Reading MA 1968.
5. Organick, E I, *Computer System Organization: The B5700/B6700 Series*, Academic Press 1973.

Designing a Command Language

G A Van den Bout
929 Tower St
Raleigh NC 27607

Nearly every system, whether it is composed of ten lines of code or ten thousand lines of code, will perform three distinct functions. It will receive input from the user, it will process this input and it will output the results. Of these three functions, the one which undoubtedly receives the least attention from the system designer is the communication from the user of the system to the system itself.

Hours and hours may be spent perfecting a processing algorithm and computing field lengths so that the resulting output can be instantly understood, yet due to the lack of consideration put into the input stage of the system, the user may be forced to plow through a series of questions and answers directed to him by the system. This is a situation which would try the patience of even the most tolerant person. Sometimes a situation even worse than this series of questions may be caused by the designer who is very familiar with the system. In an effort to save time and memory space, the designer may decide to reduce or even entirely omit any prompting by the program. This leaves the decision of what information must be entered to the intuition of the user, or to a system manual which will probably not be around when it is needed.

A good solution to the problem would be a well designed command language which would allow the user to supply all of the information which is needed by the program at one time, in a single command. Then, if any of the required data has not been entered, the computer can prompt the user for the remaining items. This method allows for both the experienced user who knows exactly what data the program needs at every instant and for the first time user who requires some help from the system now and then, but who will soon become familiar with the system and probably prefer to avoid the repetitious prompting.

Consider the following example which, although hypothetical and not necessarily typical of chess playing programs in general, illustrates problems which do exist in many systems. A superb chess playing program has been designed after months of hard work. Along with this program, a graphics output system has been devised to display the present formation of the board after each move is made. When the user sits down to test his skill against that of the machine, he becomes a partner to the following dialogue:

```
(C: COMPUTER; P: PLAYER)
C: DO YOU WISH TO MOVE(1), CAPTURE(2),
   OR CASTLE(3)? ENTER 1, 2, OR 3.
P: 1
C: ENTER NUMBER (1-8) OF ROW THAT
   PIECE IS ON.
P: 2
C: ENTER LETTER (A-Z) OF COLUMN THAT
   PIECE IS ON.
P: D
C: ENTER NUMBER (1-8) OF ROW TO WHICH
   YOU ARE MOVING.
P: . . .
```

No matter how well the machine plays chess, it is doubtful whether it will be used by any particular person for more than a few games. Despite the thought that went into the rest of the program, no creative thought was put into the command language for the system.

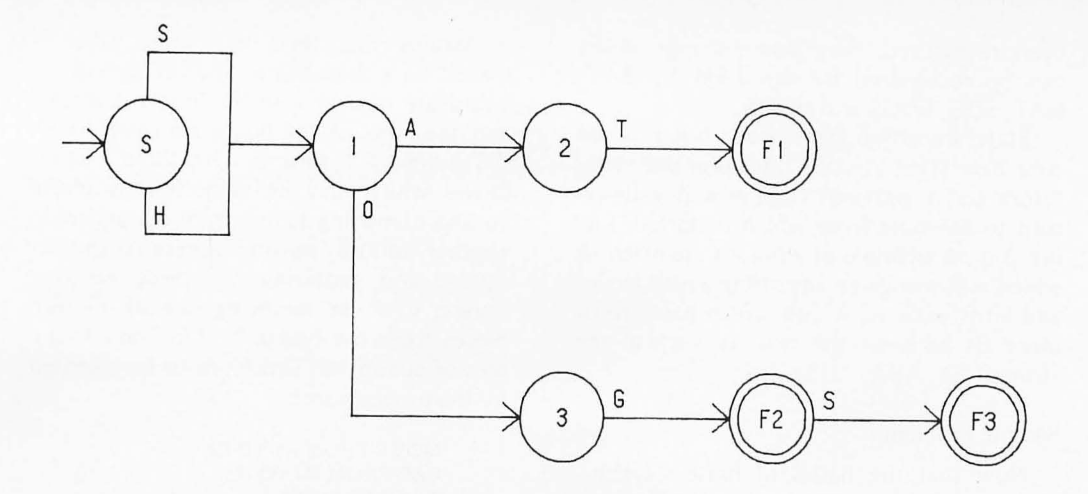


Figure 1: A finite state machine with one initial state and three final states that is capable of recognizing the words: sat, sog, sogs, hat, hog and hogs.

Now, consider the following conversation between the computer and the player.

C: ENTER YOUR FIRST MOVE.
P: MOVE FROM D2 TO D4
C: I MOVE FROM H5 TO E2. CHECK.
P: CAPTURE E2
C: FROM WHERE?
P: H2
C: . . .

This method not only cuts down on the unnecessary chatter which was encountered in the first case, but gives the player credit for possessing some knowledge of what is happening in the game. By taking time to design an easy to use command language, the designer can produce a game which will not only play well but which will also be enjoyable to use.

The problem encountered when designing a program which handles a set of commands such as these is that often no organized approach is taken to assure that the allowable commands are processed correctly. Each input string may be scanned and rescanned for the information which is needed by the program. This type of haphazard approach will very likely produce unreadable code which is hard to debug and which may contain hidden errors and ambiguities. To avoid these problems, the theory of finite state machines (FSMs) may be used to produce a recognizer program which can parse the input commands and produce a structured command which can be interpreted by the system.

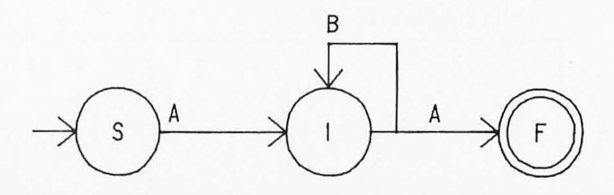
Finite State Machines

Since the aim of this article is to show how to use finite state machines to aid in programming a command language, not to thoroughly cover finite state machine theory, I will give a rather informal description of the machines. The representation used here has appeared in various places, and was chosen mainly because of its simplicity for this application.

Consider the finite state machine shown in figure 1. Each circle represents a **state** of the finite state machine. In this example there are seven states: S, 1, 2, 3, F1, F2 and F3. The names chosen for the states are arbitrary. The directed lines between the states are called **state transition paths.** The state transition path, labeled with an H, located between state S and state 1, is named S-1(H). The parenthetical symbol will be omitted when there is no ambiguity, such as the path 1-3. The states which are circled twice are **final states.** The final states in figure 1 are F1, F2 and F3. The states which are pointed to by arrows which lead from no other state are called **initial states.** The only initial state in figure 1 is S.

This finite state machine can be used to recognize several different strings, a string in this case being merely a sequence of letters. For a particular string to be recognized, an ordered path must exist between an initial state and a final state such that every symbol in the string being recognized exists (in its original order) along the path starting at the initial state. Using this finite state machine the string HOG is recognized in the following manner. Starting at initial state S, the first symbol in the string, H, leads to state 1 along path S-1(H). The second symbol, the letter O, selects path 1-3 leading to state 3. Finally, the symbol G leads to the final state F2 via the path 3-F2. Since this path exists from the initial state S to the final state F2, the string has

Figure 2: Finite state machine that has a state transition path loop.



been recognized. The other strings which can be recognized by this FSM are SAT, HAT, SOG, SOGS and HOGS.

State transition paths need not proceed to a new state. A state transition path may return to a previous state or may even return to the state from which it started. Figure 2 is an example of a finite state machine which will recognize any string which begins and ends with an A and which has zero or more Bs between the two As, such as the strings: AA, ABA, ABBA, etc.

Sample Problem

Now that the basics of finite state machines have been explained, a simple command language will be defined and implemented using them as a design tool. Using this example, a similar procedure can be followed to produce a recognizing program for nearly any command language which might be chosen.

Assume that there is a game which is played on a chess board. The columns of the board are labeled with the letters A thru H and the rows of the board are labeled with the numbers 1 thru 8. The three possible moves which may be made by any player consist of moving a piece from one square to another, MOVE, moving a piece to another square and capturing the piece on that square, CAP, or removing one of his own pieces from the board, TAKE. Some examples of commands which are to be accepted by the program are:

```
MOVE FROM A1 TO C3
CAP FROM 4H TO H1
TAKE FROM E5
MOVE TO F6 FROM 6G
```

It can be seen that the commands are made up of six basic entities which must be recognizable. Three of these entities are the commands MOVE, CAP and TAKE. TO and FROM are keywords which must be identified in order to interpret a command. The final type is a position which may consist of a letter followed by a number or a number followed by a letter and which will exist one or more times in each command.

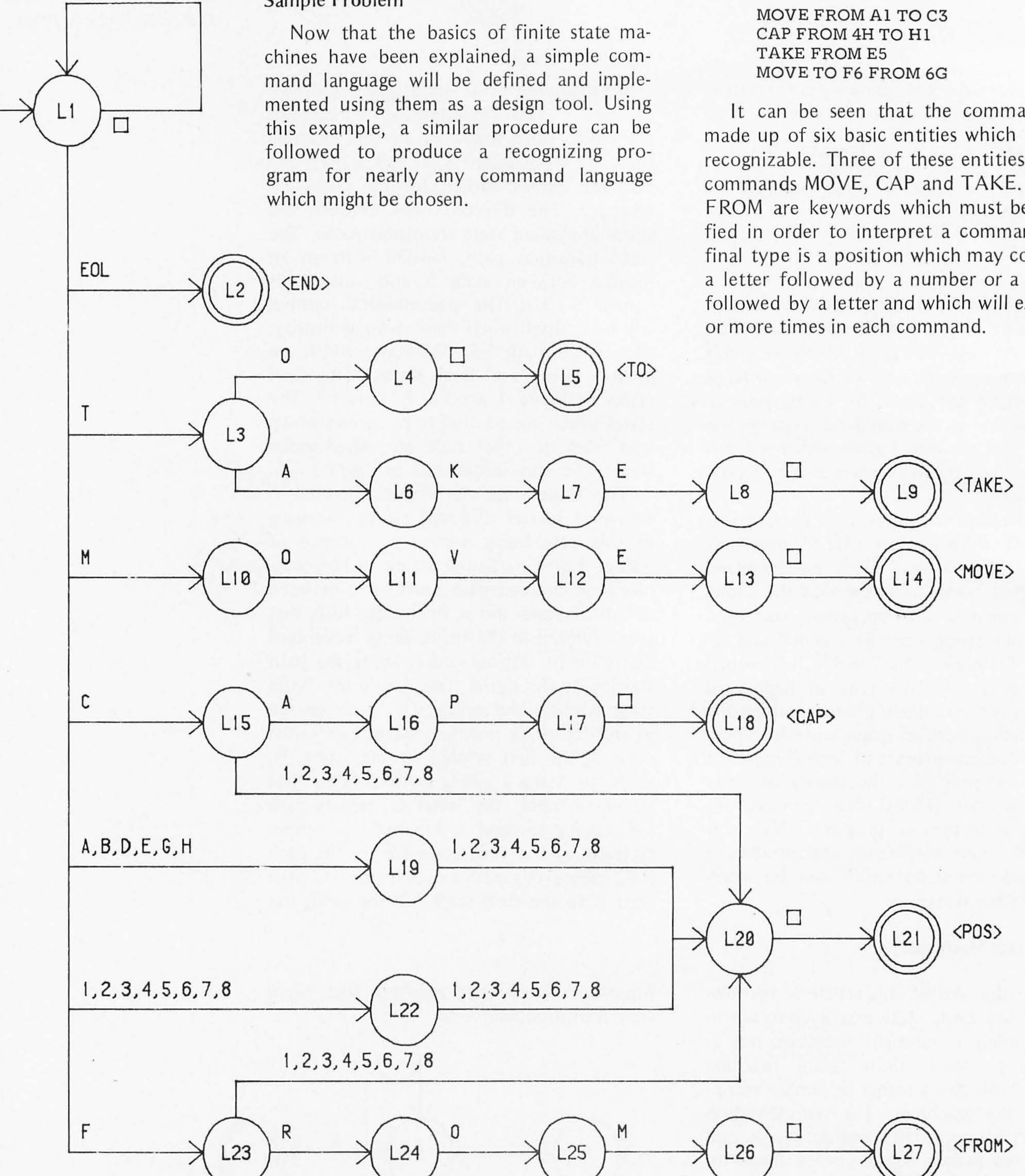


Figure 3: A lexical finite state machine for recognizing the entities that will be accepted by the game: <TO>, <TAKE>, <MOVE>, <CAP>, <FROM>, <END>, <POS>.

Command Recognizers

When a command is entered to be interpreted by the computer, it consists merely of a sequence of **symbols** (letters, numbers and spaces) which have no syntactic meaning of their own. The meaning only starts to become clear when the symbols are grouped together to form **tokens**. The tokens which exist in this game are the six entities described above. These tokens will be referred to as <MOVE>, <CAP>, <TAKE>, <TO>, <FROM>, <POS>. A finite state machine which will recognize each of these tokens is shown in figure 3. Blanks are shown on this diagram and in the following diagrams as small squares. Note that one new token has been added to the six types listed above. This new token is <END> which is recog-

S1 —<MOVE>→ S2; S1 —<CAP>→ S2; S2 —<FROM>→ S3 —<POS>→ S4 —<TO>→ S5 —<POS>→ S6; S2 —<TO>→ S8 —<POS>→ S9 —<FROM>→ S10 —<POS>→ S6; S6 —<END>→ S7; S1 —<TAKE>→ S11 —<FROM>→ S12 —<POS>→ S13 —<END>→ S14

Figure 4: A syntactic finite state machine for accepting valid commands.

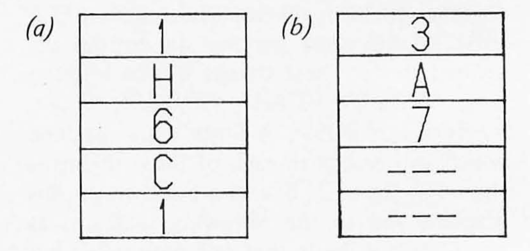


Figure 5: Two example COMMAND arrays. COMMAND array A results after processing the command MOVE TO C1 FROM H6. COMMAND array B is the result of processing TAKE FROM A7.

nized when an end of line (eol) delimiter is found.

Most of this finite state machine is self-explanatory. Note, however, the two states L15 and L23 which are entered after matching an initial C or F, respectively. These states represent a point in the matching process where the token being recognized may be either a command (<CAP> or <FROM>) or a position (<POS>). When the next symbol in the input stream is examined, the recognition of the token as a position (paths L15-L20 and L23-L20) or as a command (paths L15-L16 and L23-L24) can be made.

The finite state machine which has just been described performs the process known as **lexical analysis**, the process of grouping together input symbols to determine the tokens which have been input. The next process which must be performed is the process of **syntactic analysis**, checking the order of the tokens which have been formed to see whether they form a valid command. For example, the two "commands":

```
MOVE FROM A1 TO C3
A1 C3 FROM TO MOVE
```

are both composed of valid tokens for the example language but only the first command is syntactically correct. To determine the syntactic correctiness of a command another finite state machine must be designed. This machine, rather than having paths labeled with symbols from a character set, will have labels which are valid tokens of the language being processed. Figure 4 shows a finite state machine which will accept the valid commands of the language.

Semantic Routines

At this point two finite state machines have been produced which can be used to recognize valid commands for the game. Before these machines are used to help produce code to process actual commands, the results of processing each command must be defined. After a decision has been made regarding these results, **semantic routines**, routines to carry out the processing of the various commands, should be associated with each state transition path of the finite state machines. In our system, each command will be converted to a set of codes and placed in an array called COMMAND which will have five elements. COMMAND(1) will be set to a code describing the command operation (1=MOVE, 2=CAP, 3=TAKE), COMMAND(2) and COMMAND(3) will hold, respectively, the column and the row position associated with the FROM keyword. COMMAND(4) and COMMAND(5) will hold the column and row position associated with the TO keyword. Figure 5 shows the expected results of processing following two commands:

```
MOVE TO C1 FROM H6
TAKE FROM A7
```

For the finite state machine that is shown in figure 4, table 1 shows the semantics which will produce the desired results. Routines for paths such as S1-S2(<MOVE>) set the first element of the COMMAND array to indicate which command was recognized. Path S2-S3 is an implicit recognition of the word FROM and has no semantics associated with it since nothing must be done until the path S3-S4 is traversed. When this action occurs, the row and

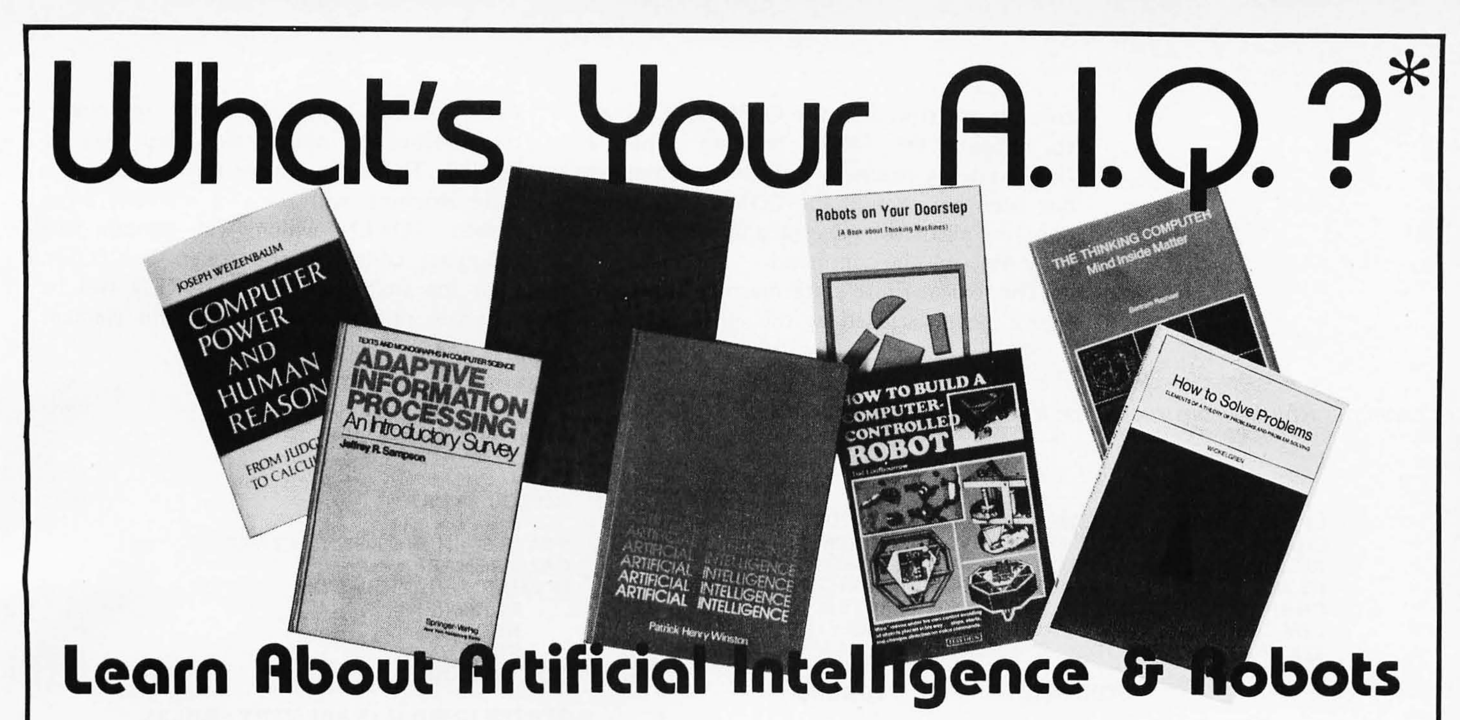
Table 1: Semantics for the syntactic finite state machine.

S1-S2(<MOVE>)	:	SET COMMAND(1)	TO 1
S1-S2(<CAP>)	:	SET COMMAND(1)	TO 2
S1-S3	:	SET COMMAND(1)	TO 3
S4-S7	:	SET COMMAND(2)	TO COLUMN (A-H)
		SET COMMAND(3)	TO ROW (1-8)
S10-S13	:	SET COMMAND(4)	TO COLUMN (A-H)
		SET COMMAND(5)	TO ROW (1-8)
S8-S9	:	SET COMMAND(4)	TO COLUMN (A-H)
		SET COMMAND(5)	TO ROW (1-8)
S10-S6	:	SET COMMAND(2)	TO COLUMN (A-H)
		SET COMMAND(3)	TO ROW (1-8)
S12-S13	:	SET COMMAND(2)	TO COLUMN (A-H)
		SET COMMAND(3)	TO ROW (1-8)
OTHERS	:	(NO SEMANTICS)	

Table 2: Semantics for the lexical finite state machine. These routines are used to set up the array TOKEN.

L1-L2	:	SET TOKEN(1)	TO 0
		SET TOKEN(2)	TO 6
L4-L5	:	SET TOKEN(1)	TO 0
		SET TOKEN(2)	TO 4
L8-L9	:	SET TOKEN(1)	TO 0
		SET TOKEN(2)	TO 3
L13-L14	:	SET TOKEN(1)	TO 0
		SET TOKEN(2)	TO 1
L17-L18	:	SET TOKEN(1)	TO 0
		SET TOKEN(2)	TO 2
L26-L27	:	SET TOKEN(1)	TO 0
		SET TOKEN(2)	TO 5
L1-L19	:	SET TOKEN(2)	TO INPUT CHARACTER
L1-L22	:	SET TOKEN(1)	TO INPUT CHARACTER
L19-L20	:	SET TOKEN(1)	TO INPUT CHARACTER
L22-L20	:	SET TOKEN(2)	TO INPUT CHARACTER
L15-L20	:	SET TOKEN(1)	TO INPUT CHARACTER
		SET TOKEN(2)	TO "C"
L23-L20	:	SET TOKEN(1)	TO INPUT CHARACTER
		SET TOKEN(2)	TO "F"
OTHERS	:	(NO SEMANTICS)	

What's Your A.I.Q.?*



Learn About Artificial Intelligence & Robots

INTRODUCTION TO ARTIFICIAL INTELLIGENCE by Philip C Jackson Jr.

□ This book surveys the field beginning with Turing's test, the mathematical description of phenomena, finite state machines and limits to computational ability, followed by chapters on problem solving, game playing, pattern perception, theorem proving, semantic information processing, parallel processing, evolutionary systems, robots, and a look at the future of the field. This thoughtful and unusual book will make a useful addition to your library. A 50-page bibliography is included. 453 pp. $18.50 hardcover (no photo).

ARTIFICIAL INTELLIGENCE by Patrick Henry Winston.

□ This book is designed for use in a course on artificial intelligence, but should prove invaluable to the newcomer to the topic, as well as being useful to the experienced as a reference text. Part One is set up to be used as an introduction to the key ideas of the subject, including the basics of elementary representation, symbolic constraint exploitation, search, and control. In addition, understanding natural language, representing common sense meaning, computer vision, and expert problem solving are covered. Part Two covers the programming language LISP, its organization and uses. Profusely illustrated. Only $17.95 hardcover. 444 pp.

ADAPTIVE INFORMATION PROCESSING, AN INTRODUCTORY SURVEY by Jeffrey R Sampson.

□ Even the most complex of today's computers is dwarfed by the capabilities of its designer's brain. Imitation of the capabilities and adaptive nature of biological information processing systems has long been a goal of computer designers. Jeffrey R Sampson presents an introductory survey of this field. From the fundamentals of information and information processing entities, he examines the richness of biological information processing and goes on to cover the application of computers to artificial intelligence. His is a view which unifies the diverse aspects of the subject of artificial intelligence, emphasizing adaptive behavior. 214 pp. $14.80 hardcover.

COMPUTER MODELS OF THOUGHT AND LANGUAGE edited by Roger C Schank & Kenneth Mark Colby.

□ Part of the current work in AI involves the modeling of human mental processes in computer programs. Here is a collection of articles, which deal with natural language models, models of belief systems, and models of memory and cognition. Required reading for AI enthusiasts. 450 pp. $24.95 hardcover.

HOW TO BUILD A COMPUTER-CONTROLLED ROBOT by Tod Loofbourrow.

□ This book combines the dream of robotics—to create an intelligence other than human—with the reality, by providing both hands-on experience with robotics and an application of a microprocessor. It details the step-by-step directions for building a robot, named "Mike," controlled by a KIM-1 microprocessor, with the complete control programs clearly written out. 132 pp. $7.95.

ROBOTS ON YOUR DOORSTEP by Nels Winkless and Iben Browning.

□ "This book will amaze you, frighten you, nauseate you, excite you. . .it will probably make you think about things you have never contemplated before," states the introduction to this clever and well-written account of robots: past, present, and future. Intelligence, artificial intelligence, brain structure and simulation, and characteristics of robots are only a few of the areas explored. 179 pp. $6.95.

HOW TO SOLVE PROBLEMS by Wayne A Wickelgren.

□ Analyzes and systematizes the basic methods of solving mathematical problems. The methods are described in terms of a modern theory derived from research in computer simulation of thinking. Examples illustrating these methods include chess problems, logical puzzles, and railroad switching problems frequently encountered in science and engineering. $6.50.

THE THINKING COMPUTER: MIND INSIDE MATTER by Bertram Raphael.

□ With a minimum of technical jargon, this book discusses the capabilities of modern digital computers and how they are being used in contemporary AI research. It discusses the progress of AI, the goals, and the variety of current approaches to making the computer more intelligent. $8.50

COMPUTER POWER AND HUMAN REASON by Joseph Weizenbaum.

□ If you're presently a programmer by trade or skill, you'll see a philosophy of computer use and abuse propounded. If you're a novice, the tutorial and explanatory chapters of this book serve as an excellent and eminently readable background source. This includes an outstanding, low level explanation of what an algorithm is, and how computers execute effective algorithms. $6.50

BITS™ inc Books to erase the impossible

POB 428, 25 Route 101 West, Peterborough NH, 03458

For your convenience in ordering, please use this page plus the order form on page 200. You may photocopy this page.

DIAL YOUR CHARGE CARD ORDERS TOLL FREE
(800) 258-5477. Mon - Fri 9 - 5 PM

**Artificial Intelligence Quotient*

column are stored in the COMMAND array to indicate the FROM position. When a final state is reached, an entire command has been parsed and the COMMAND array contains all of the necessary information to fully describe the command.

The lexical finite state machine shown in figure 3 will be used by the syntactic finite state machine just described to obtain tokens from the input stream when they are needed. The output from the lexical finite state machine will be a 2 element array named TOKEN which will contain the following codes. If the token is <POS>, then the first element of TOKEN will be the row number and the second element

Listing 1: Routine constructed for the lexical finite state machine.

```
*
*        LEX IS A SUBROUTINE WHICH EXAMINES INPUT
*        CHARACTERS UNTIL IT FINDS A VALID TOKEN OR
*        AN INPUT ERROR. SUBROUTINE RCHAR READS THE
*        NEXT CHARACTER FROM THE INPUT BUFFER INTO
*        CHAR. '#' IS THE END-OF-BUFFER CHARACTER.
*        LEX SETS TOKEN (THE TWO ELEMENT ARRAY) TO
*        THE FOLLOWING CODES:
*
*                          TOKEN(1)         TOKEN(2)
*
*        <MOVE>   -          0                 1
*        <CAP>    -          0                 2
*        <TAKE>   -          0                 3
*        <TO>     -          0                 4
*        <FROM>   -          0                 5
*        <END>    -          0                 6
*        ERROR    -          0                 7
*        <POS>    -       ROW: 1-8          COL: A-Z
*
LEX:     SUBROUTINE;
         TOKEN(1) = 0
*
*        STATE 1 - BEGINNING STATE
L1:      CALL RCHAR( );
         IF CHAR = ' ' THEN GO TO L1;
         IF CHAR = 'T' THEN GO TO L3;
         IF CHAR = 'M' THEN GO TO L10;
         IF CHAR = 'C' THEN GO TO L15;
         IF CHAR = 'F' THEN GO TO L23;
         IF CHAR = '#' THEN DO;
              TOKEN(2) = 6;
              RETURN;
              END;
         IF CHAR = 'A' | 'B' | 'D' | 'E' | 'G' |
                   'H' THEN DO;
              TOKEN(2) = CHAR;
              GO TO L19;
              END;
         IF CHAR = '1' | '2' | '3' | '4' | '5' |
                   '6' | '7' | '8' THEN DO;
              TOKEN(1) = CHAR;
              GO TO L22;
              END;
         GO TO LEXERR;
*
*        STATE 3 - HAVE FOUND 'T'
L3:      CALL RCHAR( );
         IF CHAR = 'O' THEN GO TO L4;
         IF CHAR = 'A' THEN GO TO L6;
         GO TO LEXERR;
*
*        STATE 4 - HAVE FOUND <TO>
L4:      CALL RCHAR( );
         IF CHAR = ' ' THEN DO;
              TOKEN(2) = 4;
              RETURN;
              END;
         GO TO LEXERR;
*
*        STATE 6 - HAVE FOUND 'TA'
L6:      CALL RCHAR( );
         IF CHAR = 'K' THEN GO TO L7;
         GO TO LEXERR;
*
*        STATE 7 - HAVE FOUND 'TAK'
L7:      CALL RCHAR( );
         IF CHAR = 'E' THEN GO TO L8;
         GO TO LEXERR;
*
*        STATE 8 - HAVE FOUND <TAKE>
L8:      CALL RCHAR( );
         IF CHAR = ' ' THEN DO;
              TOKEN(2) = 3;
              RETURN;
              END;
         GO TO LEXERR;
*
*        STATES 10 THRU 13 ARE VERY SIMILAR
*        TO STATES 3 THRU 8 ABOVE AND ARE
*        NOT SHOWN.
*
*        STATE 15 - HAVE FOUND 'C'
L15:     CALL RCHAR( );
         IF CHAR = '1' | '2' | '3' | '4' | '5' |
                   '6' | '7' | '8' THEN DO;
              TOKEN(1) = CHAR;
              TOKEN(2) = 'C';
              GO TO L20;
              END;
         IF CHAR = 'A' THEN GO TO L16;
         GO TO LEXERR;
*
*        STATES 16 AND 17 RECOGNIZE THE REST OF
*        <CAP> AND ARE NOT SHOWN.
*
*        STATE 19 - HAVE FOUND COLUMN LETTER (A-Z)
L19:     IF CHAR = '1' | '2' | '3' | '4' | '5' |
                   '6' | '7' | '8' THEN DO;
              TOKEN(1) = CHAR;
              GO TO L20;
              END;
         GO TO LEXERR;
*
*        STATE 20 - HAVE FOUND <POS>
L20:     IF CHAR = ' ' THEN RETURN;
         GO TO LEXERR;
*
*        STATE 22 - HAVE FOUND ROW NUMBER (1-8)
L22:     IF CHAR = 'A' | 'B' | 'C' | 'D' | 'E' |
                   'F' | 'G' | 'H' THEN DO;
              TOKEN(2) = CHAR;
              GO TO L20;
              END;
         GO TO LEXERR;
*
*        STATE 23 - HAVE FOUND 'F'
L23:     IF CHAR = '1' | '2' | '3' | '4' | '5' |
                   '6' | '7' | '8' THEN DO;
              TOKEN(1) = CHAR;
              TOKEN(2) = 'F';
              GO TO L20;
              END;
         IF CHAR = 'R' THEN GO TO L24;
         GO TO LEXERR;
*
*        STATES 24 THRU 26 ARE SIMILAR TO OTHER
*        STATES WHICH RECOGNIZE KEYWORDS AND ARE
*        NOT SHOWN.
*
*        LEXERR - AN ERROR HAS BEEN ENCOUNTERED
*                 IN THE INPUT STRING.
LEXERR:  TOKEN(1) = 0;
         TOKEN(2) = 7;
         RETURN;
END LEX;
```

Listing 2: Routine constructed for the syntactical finite state machine.

```
*
*           SYN IS A SUBROUTINE WHICH EXAMINES INPUT
*           TOKENS TO DETERMINE IF A COMMAND IS OR IS
*           NOT VALID. SYN USES SUBROUTINE LEX TO
*           OBTAIN THE TOKENS FROM THE INPUT STREAM.
*           A FIVE ELEMENT ARRAY NAMED COMMAND IS
*           SET USING THE FOLLOWING CODES:
*
*           COMMAND(1)  :  0=ERROR, 1=MOVE, 2=CAP, 3=TAKE.
*           COMMAND(2)  :  COLUMN (A-H) OF "FROM".
*           COMMAND(3)  :  ROW (1-8) OF "FROM".
*           COMMAND(4)  :  COLUMN (A-H) OF "TO".
*           COMMAND(5)  :  ROW (1-8) OF "TO".
*
SYN:        SUBROUTINE;
*
*           STATE 1 - BEGINNING STATE
S1:         CALL LEX( );
            IF TOKEN(1)=0 & TOKEN(2)=1 THEN DO;
                COMMAND(1) = 1;
                GO TO S2;
                END;
            IF TOKEN(1)=0 & TOKEN(2)=2 THEN DO;
                COMMAND(1) = 2;
                GO TO S2;
                END;
            IF TOKEN(1)=0 & TOKEN(2)=3 THEN DO;
                COMMAND(1) = 3;
                GO TO S3;
                END;
            GO TO SYNERR;
*
*           STATE 2 - <MOVE> OR <CAP> FOUND
S2:         CALL LEX( );
            IF TOKEN(1)=0 & TOKEN(2)=5 THEN GO TO S3;
            IF TOKEN(1)=0 & TOKEN(2)=4 THEN GO TO S4;
            GO TO SYNERR;
*
*           STATE 3 - <MOVE><FROM> FOUND
S3:         CALL LEX( );
            IF TOKEN(1)>0 THEN DO;
                COMMAND(2) = TOKEN(2);
                COMMAND(3) = TOKEN(1);
                GO TO S4;
                END;
            GO TO SYNERR;
*
*           STATE 4 - <MOVE><FROM><POS> FOUND
S4:         CALL LEX( );
            IF TOKEN(1)=0 & TOKEN(2)=4 THEN GO TO S5;
            GO TO SYNERR;
*
*           STATE 5 - <MOVE><FROM><POS><TO> FOUND
S5:         CALL LEX( );
            IF TOKEN(1)>0 THEN DO;
                COMMAND(4) = TOKEN(2);
                COMMAND(5) = TOKEN(1);
                GO TO S6;
                END;
            GO TO SYNERR;
*
*           STATE 6 - ENTIRE COMMAND FOUND
S6:         CALL LEX( );
            IF TOKEN(1)=0 & TOKEN(2)=6 THEN RETURN;
            GO TO SYNERR;
*
*           STATES 8 THRU 13 ARE VERY SIMILAR TO STATES
*           2 THRU 6 AND ARE NOT SHOWN.
*
*           SYNERR - INVALID COMMAND SYNTAX.
SYNERR:     COMMAND(1) = 0;
            RETURN;
END SYN;
```

will be the column letter. If the token is not <POS>, then the first element of TOKEN array will be set to zero and the second element will be a code indicating which type of token was recognized (1 for <MOVE>, 2 for <CAP>, 3 for <TAKE>, 4 for <TO>, 5 for <FROM>, 6 for <END>). The semantic routines associated with the lexical finite state machine to set TOKEN correctly are shown in table 2.

Implementation

The first step in implementing the command language is the conversion of the lexical finite state machine into a subroutine which locates the next token in the input stream and places the necessary codes into TOKEN as described above. If at any time, an error is detected while attempting to recognize a new token from the input stream, then TOKEN(1) is set to zero, TOKEN(2) is set to 7 and this routine returns to its calling routine.

A program named LEX, written in a BASIC-like language, which accomplishes these results is shown in listing 1. Prior to the invocation of this routine, the input command must be obtained from the user and stored in a buffer followed by a blank and the end of line character. A routine RCHAR is assumed to exist, which reads the next character from the input buffer and places it into the variable CHAR. Because of the way that the program has been designed, the flow of the program is easy to understand and modifications are easy to make if necessary, especially if the corresponding finite state machine diagram is available. The program is divided into sections which correspond to the states in the finite state machine. Each section determines which state transition pointer should be followed from the character which is being scanned. It then performs the semantics associated with this state transition pointer and moves along the path by means of the appropriate GOTO statement. If during the processing of any state, the input character being examined does not correspond with any valid state transition pointer, the routine sets TOKEN to the error code described above and returns to its caller.

Listing 2 shows the routine constructed from the syntactic finite state machine. The

structure of this program is almost identical to the structure of the previous routine. This time each section of the program examines the next token which has been obtained by a call to LEX, performs the appropriate semantics for the path to be traversed, and then moves to the next defined state. Again, if either an invalid token is encountered or if the routine LEX returns an error code, this routine returns to its caller after leaving an error code of zero in COMMAND.

Due to the way these routines were constructed, a single error code is returned if any error occurs in a command. But, because the exact location in the state diagram is known whenever an error occurs, more descriptive error messages can be generated, or fix up action may be performed. If the command:

MOVE TO A8

is entered, then the syntactic routine would encounter the <END> token while processing state S8. Based on the present form of the program, the error message printed would most likely be "INVALID COMMAND SYNTAX – ENTER NEW COMMAND" since no attempt is made to analyze the syntax error.

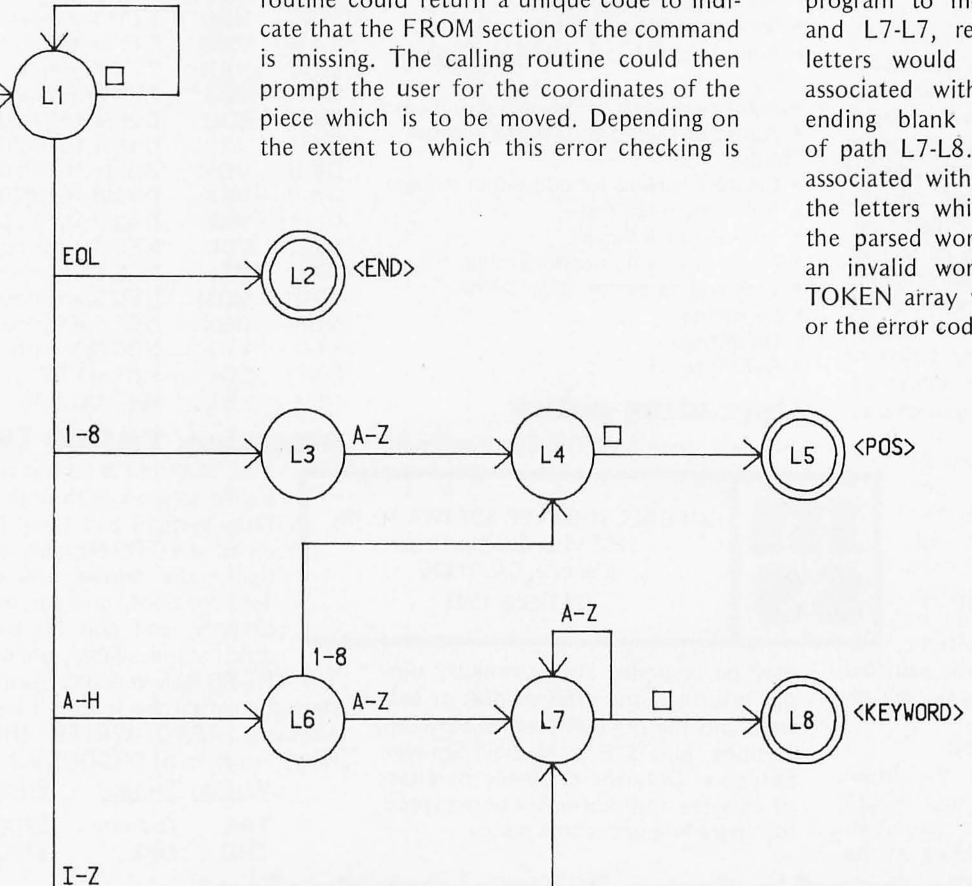
However, instead of merely returning the zero error code to its caller, the syntactic routine could return a unique code to indicate that the FROM section of the command is missing. The calling routine could then prompt the user for the coordinates of the piece which is to be moved. Depending on the extent to which this error checking is carried out, a very elaborate and easy to use command system can be created.

Other Representations

The finite state machine diagrams in figures 3 and 4 have been chosen to illustrate the techniques of using finite state machines for designing command languages and do not represent the only way to implement this sample command language. An alternate finite state machine which performs lexical analysis for the example game is shown in figure 6. In this finite state machine all of the commands and keywords (MOVE, CAP, TAKE, TO and FROM) map into the single token <KEYWORD>. Semantic routines associated with the paths L1-L6, L1-L7, L6-L7 and L7-L7 would be used to save the symbols which have already been matched. Then when path L7-L8 is traversed, the semantics associated with this path would include a table lookup routine to identify the command or keyword and correctly fill in the TOKEN array.

To illustrate this technique, observe how the finite state machine in figure 6 would recognize the capture command. Starting with state L1, the C would cause the traversal of path L1-L6 and would be saved to later help identify the token being parsed. The A and the P would similarly cause the program to move along the paths L6-L7 and L7-L7, respectively, and again these letters would be saved by the semantics associated with these paths. Finally, the ending blank would cause the traversal of path L7-L8. At this time, the semantics associated with path L7-L8 would examine the letters which had been saved, identify the parsed word as either a valid token or an invalid word, and correctly fill in the TOKEN array with the code for the token or the error code.

Figure 6: An alternate solution for the lexical analysis of the game program.



Certain advantages exist for both the method used in the finite state machine in figure 3 and for this method but as the number of keywords increases, this method becomes much more efficient in terms of memory used.

Conclusion

The purpose of this article has been to show how finite state machine theory may be applied to produce correct and well structured code for command recognizers. I have used finite state machines to produce both an information retrieval command language and a FORTRAN free format input processor of character strings and numbers; and methods similar to these shown here have significantly speeded up the implementations. The efficiency of this method will vary depending on which language is used to program the procedures and on the programming techniques used. The sample programs previously shown were designed with clarity in mind and are not the most efficient routines which could have been written. I would recommend that the lexical finite state machine be coded in assembler language if possible since many techniques exist to improve the performance of character by character scanning and comparison. Of course, both of the routines may be written in any language desired, but because of the memory space limitations of most small computers, assembler language would probably be an asset. As memory size increases, however, the advantages of assembler tend to decrease. Whichever language is chosen, the finite state machine method of designing a command language should produce a system which runs correctly after less programming effort, which can be more readily understood and changed as necessary, and which can provide a series of error and prompting messages that help to make the system easier and more enjoyable to use.■

REFERENCES

For examples of the use of finite state machines to identify tokens of a programming language I refer the reader to the following:

Gries, David, "The Scanner," *Compiler Construction for Digital Computers*, John Wiley and Sons, New York, 1971, pages 64 thru 71.

More information on finite state machines and their theory can be found in many other books, including:

Gill, A, *Introduction to the Theory of Finite State Machines,* McGraw-Hill, New York, 1962.

Technical Forum

Approximation Makes a Magnitude of Difference

Bob Leedom
14069 Stevens Valley Ct
Glenwood MD 21738

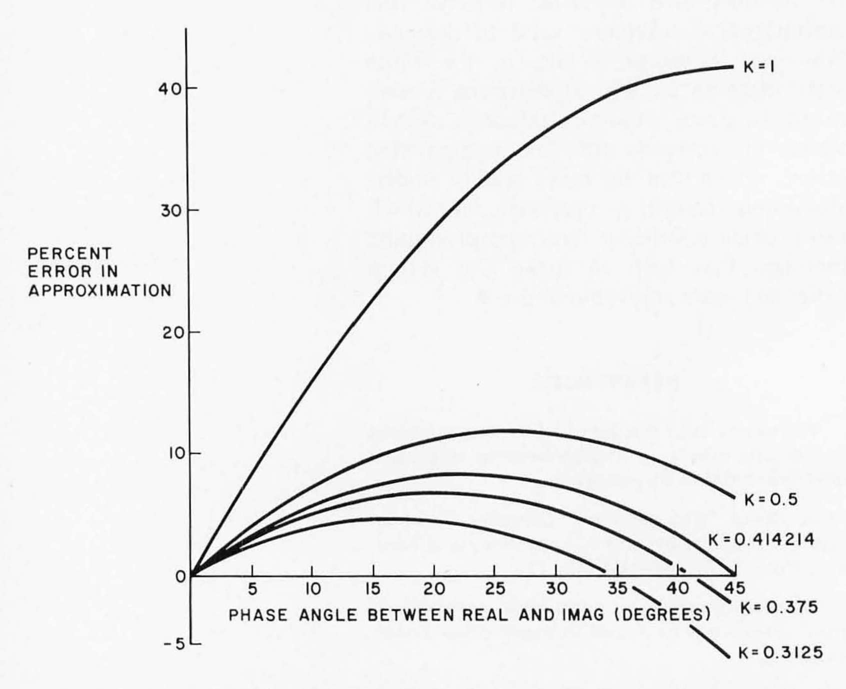


Figure 1: A plot of the percent error in the magnitude approximation for different values of K. We are approximating the square root of $REAL^2 + IMAG^2$ by the formula L + KS, where L is the larger and S is the smaller value of the absolute values of the real and imaginary parts of a vector.

K		PEAK ERRORS VARY FROM (%)	RMS ERROR (%)
1		0 to 41	30
.5	(1/2)	0 to 12	9.1
.414214	($\sqrt{2} - 1$)	0 to 8.2	5.9
.375	(3/8)	−2.8 to 6.8	4.7
.3125	(5/16)	−7.2 to 4.8	3.6

Table 1: The accuracy of the approximation algorithm as a function of K.

K		IMPLEMENTATION OF K x ABS (SMALLER)	
1		None Required.	(Fastest)
.5	(1/2)	Shift right 1.	
.375	(3/8)	Shift right 2, Store in TEMP, Shift right 1, Add TEMP.	
.3125	(5/16)	Shift right 2, Store in TEMP, Shift right 2, Add TEMP.	
.414214	($\sqrt{2} - 1$)	Multiply.	(Slowest)

Table 2: A comparison of implementation speeds for various values of K.

I enjoyed Richard Lord's article presenting an assembly language FFT (fast Fourier transform) program for the 6800 (February 1979 BYTE, page 108). Adaptation to my 6502 (KIM) system should be fairly straightforward.

However, the author notes that obtaining the magnitude of each resulting vector is almost as time-consuming as the FFT process itself, since this would involve taking the square root of the sum of the squares of each REAL/IMAG pair. Strictly speaking, he is correct, but with very little trouble a quite reasonable approximation to the correct magnitude can be found. The following algorithm is often used for this purpose in the processing of speech and radar data, and may be implemented easily in either hardware or software.

To find the magnitude of a vector, given the orthogonal components (eg: REAL and IMAG):

- take the absolute values of REAL and IMAG;
- compare the two absolute values, place the larger in L and the smaller in S — if they're equal, it doesn't matter which goes where;
- multiply S by a constant (K), add the result to L.

What is K? That depends on how much accuracy you're willing to sacrifice for computation speed. To appreciate this, you should understand that the error in the magnitude computation will be a function of the phase angle between the two components. In his article, Mr Lord simply added L to S, thus letting K = 1. This approximation gives an error of from 0 to 41 percent:

Let MAGN = L + S.

Suppose a vector actually has a magnitude of 100 units. If L = 0 and S = 100, then MAGN = 100, or 0 percent error. But, if L = 70.7 and S = 70.7, then MAGN = 141.4, or 41.4 percent error.

Table 1 shows several values of K, along with the corresponding spread of the peak errors. However, just looking at the peak errors can be deceiving; what you really

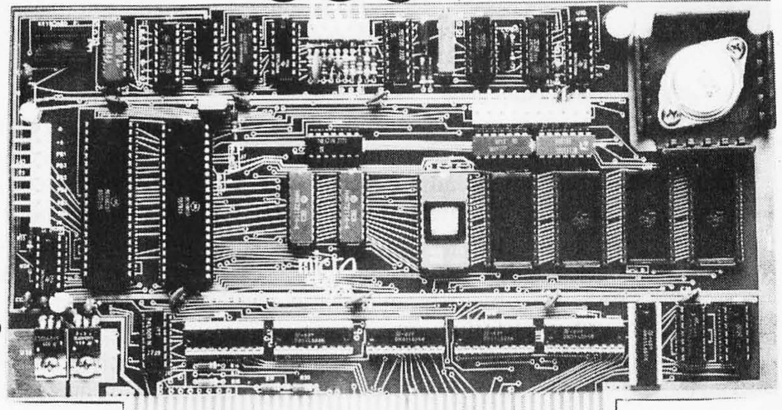
want to do is minimize some measure of the average error. Since the error function "folds" at 45 degrees of phase angle between L and S, I wrote a short program to compute the error at 1 degree intervals from 0 to 45 degrees. The root mean square of these errors is given in table 1 as a sort of quality factor for a given value of K.

As you can see, the computation of the magnitude can be improved from 3 to 8 times, simply by choosing the appropriate value of K. The error reduction as a function of K is shown graphically in figure 1; this is the accuracy part of the tradeoff.

The other side of the coin is speed of implementation. Given the absolute values of REAL and IMAG, and the fact that some fraction of one will be added to the other, it takes no extra time to perform the algorithm with K=1. However, with K=.414214, you must multiply (after finding the smaller of the two absolute values). The range of in-between speeds is given in table 2.

The accuracy/speed tradeoff should be evaluated for each user's application, either by analysis or by trying possible values of K. However, if you don't have the time or energy for this, remember that an immediate reduction of the root mean square error to less than ten percent may be obtained by a compare and a shift (ie, K = .5). ■





Concerning User's Manuals

H Edgar Coburn
Structural Engineer
3560 Albatross St
San Diego CA 92103

In my opinion, many engineers are incompetent when it comes to transmitting information to anyone not already trained in their particular specialty. (I, myself, am wondering if I'm getting *this* message across.) Engineers, who generally are not human oriented, excuse the obscurity of their communications by charging that the other person (*not* one of their compatriots, obviously) is not too bright. Unfortunately, there is just enough truth in this assumption to convince most engineers that there is no need to undertake the drudgery of learning how to be lucid — nothing is so easy as adjusting facts to fit opinions.

It is clearly apparent that if engineers are no more obscure in discourse than other people, then I don't have much of a point. So, before plunging into the real subject of this article, let's examine this question with respect to one particular feature involved in the process of transmitting information to others — namely, indices. Information that is not accessible, or that is accessible only with excessive difficulty, is not of much practical value.

In connection with another project, I made a statistical study of the indices of the nonfiction books available in a large public library. I found that the average amount of space devoted to indexing was approximately 1.8% of the total number of pages in a book. Indices varied in size from 0 to more than 7% of the book pages. While it is evidently true that index length is no measure of index quality, it is equally apparent that a short index is limited in the amount of information that it can transmit.

Engineering books, despite the complexity of their subject matter, have less indexing (at an average of 1.3%) than nonfiction books in general. On the other hand, science books, properly reflecting the complexity of their subject matter, have more indexing (at an average of 2.4%) than nonfiction books in general.

Unfortunately, many instruction manuals for computers have been written by engineers. It may well be that the obscurity of computer manuals has a substantial effect on personal computer sales. It is even conceivable that literally thousands of intelligent, educated people, those who might benefit from the possession of a personal computer, are "turned off" when they see

some of the instruction manuals published by computer manufacturers. If the reader is skeptical on this point, let him compare the average personal computer user's manual with a really good manual, such as the one supplied with Hewlett-Packard's HP-67 calculator.

Suppose that in a given year 10,000 people are turned away from personal computers by the paucity of lucid manuals. (I'm safe here because no one really knows how many potential buyers don't buy.) And suppose that the average user investment in personal computers is (optimistically) $4000. This means that the personal computer industry, with at least a few manufacturers and retail sellers on the ragged edge of solvency, may be needlessly driving no less than $40,000,000 per year into other hands.

If all computer manufacturers were to test their manuals by having several intelligent, educated people try to operate the corresponding computer with nothing but the manuals for a guide, the results might be illuminating, or even startling to the manufacturers. Then, instead of assuming that these test users are clumsy, it might be helpful to revise the manuals until they are lucid, not invincibly obscure.

I believe that the manufacturers will discover, if they actually make such a test, that the choice of a specific word is highly significant in the transmission of information. It makes a heap of difference whether one says, "Woman *and* child" or "Woman *with* child." It also makes a difference whether an engineer writes, ". . . *has* a directory entry" or ". . . *requires* a directory entry."

But most significant of all, many engineers seem to think that the reader needs little or nothing in the way of orientation. This, unfortunately, is not true. The reader of a manual needs to be led by the hand all the way — good writing typically provides such assistance. What often happens in practice is that the reader is given the brush-off, with the declaration that the manual assumes that the reader is acquainted with the subject. This is a luxury that the personal computer industry cannot afford.

Of course, it is perfectly reasonable to assume that the expectant owner of a personal computer should do a little studying of BASIC, for instance. The user definitely won't be harmed if he or she gets some idea of the general organization of a computer. But there is grave doubt that such training will aid the user noticeably in learning how, for example, to manage the disk file of some particular computer.

Furthermore, such learning will be of no value at all in helping the hopeful beginning reader in remembering the casual comment, appearing many pages earlier, which mentions that control C is necessary whenever "2A00 hex" is transposed with M5700 in the . . . oh well, what does it matter? The point is that the novice computer owner needs guidance. Among other things, this means *examples, examples,* and more *examples.* Since the personal computer industry has not been overwhelmed by standardization, even the experienced computer user needs lucid, particular instructions when adapting to a new system.

One way to improve the situation is to have manuals written by those few engineers who have demonstrated a knack for putting ideas across. And since behavioral phenomena are complex, it is even better to engage the talents and specialized training of a psychologist. It is obvious to a psychologist that students need orientation and examples in order to develop skill in a reasonable time.

Unfortunately, it's one thing to observe a need and quite another to get action. I don't imagine for a moment that it is possible to get any action out of people without some kind of motivation. However, it should be noted that motivation techniques, like everything else, have changed remarkably over the years. In the 16th century, for instance, it was fairly common practice when a king was offended by some luckless peasant (or even by a nobleman) to have the offender bodily pulled apart by teams of Clydesdale horses (the kind that pull beer wagons). This sort of example was supposed to ensure a certain amount of respect for His Majesty. It was motivation that everyone could understand. Sadly, we're so civilized now that we can't use any of the old-time, sure-cure methods of motivation.

If engineers knew that the penalty for failure to be lucid was to be pulled apart by teams of draft horses, it might have a salutary effect on the writings of engineers. (If some computer engineers now think that I should be pulled apart by teams of draft horses, it clearly indicates that my writings are lucid. The readers get the message. Therefore I should *not* be pulled apart. Q.E.D.)

If a user's manual confuses intelligent people, it is not only unsatisfactory to the user, but damages the fortunes of manufacturers and retail dealers also, because poor documentation inhibits sales. Why should the manufacturers pay for full page color advertisements featuring their products, only to throw the benefits away by offering obscurely written manuals? There must be a better way.■

Double Sided Notes

Jonathan A Titus
TYCHON Inc
POB 242
Blacksburg VA 24060

David Lamkins' article about printed circuit layout techniques "Designing With Double Sided Printed Circuit Boards" (March 1979 BYTE, page 94) described some techniques that shouldn't be used in good printed circuit board designs. The main problem is the *strategy* of designing the power and ground runs as the last step. Don't do it.

The power and ground runs should be designed first in the printed circuit board layout, and not last. Here is why.

- Power runs should be as wide as possible. It is difficult to make them very wide if you have to make them fit between signal runs, pins, etc.
- It will be almost impossible to add decoupling capacitors to power runs that snake through signal runs. Remember, you will need one decoupling capacitor per 7400 series integrated circuit in a good design.
- If the power runs are left until last, poor design takes over, making the designer seek ground and power connections wherever they are available. Potential differences often occur, and the circuit doesn't function. Ground loops are also a problem unless proper layout of power runs is observed early in the design.

Although the use of one colored pencil per side of the double-sided printed circuit board is noted, designers should try to keep the runs on one side oriented in a right-left fashion and those on the other side oriented in an up-down fashion. If this course is followed, problems such as those in Lamkins' figure 4 are avoided. Use of this technique also simplifies *problems* such as those shown in Lamkins' figure 3 design.

The article also mentions the use of a *dedicated through-hole*. This is a new one to me. I always thought that the holes on a printed circuit board were dedicated to something. If they weren't *dedicated*, there wouldn't be any need for them. What is a through-hole? I thought that holes went through something. Who knows, maybe some computerist will come up with a nested-hole; a small hole inside a bigger one. Best wishes.■

Statistical Computations Recomputed

J G Bliss
2141 Cumberland Av S
Saskatoon, Saskatchewan
CANADA S7J 1Z2

Alan B Forsythe, in his article "Elements of Statistical Computation," (January 1979 BYTE, page 182) states:

> Several books of BASIC programs include the calculation of the standard deviation. Those I checked give the wrong answer for this set of data.

This is probably a result of the formula used in the article:

$$s = \sqrt{\frac{1}{N-1} \Sigma (X - \overline{X})^2} \ .$$

The usual formula for standard deviation is:

$$\sigma = \sqrt{\frac{1}{N} \Sigma (X - \overline{X})^2} \ .$$

The version given in the article is used when calculating standard deviation from a sample. (See for example *Handbook of Sampling for Accounting and Auditing*, 2nd edition, by Herbert Arkin, McGraw-Hill.)■



Alan Kaniss
Vincent DiChristofaro
John Santini
1327 McKinley St
Philadelphia PA 19111

The Great APL Contest –

After many trials and tribulations, we are happy to announce the winners of the *Great APL Contest* (August 1977 BYTE). The object of the contest was to create a usable APL interpreter.

The first place prize of $1000 went to Alan Kaniss, Vincent DiChristofaro, and John Santini for their APL interpreter written in Pascal. This was the most complete interpreter we received.

The second place prize of $500 went to two groups: the APL *Committee of* Texas A&M Microcomputer Club which submitted a club entry, and Stephanie Charles and Normand Berube who submitted a jointly written program. Both of these programs were for 8080 processor machines.

We thank all the people who entered the contest for the time they spent writing their interpreters, and we hope that they learned a great deal from the experience.

We used Michael Wimble's flowcharts (see "An APL Interpreter for Microcomputers," August, September, October 1977 BYTE) as generalized guidelines for our APL interpreter, rather than coding directly from them. We used most of his ideas on function implementation, table storage, input scanning, and statement parsing. There were a few minor errors in logic, but for the most part the flowcharts were clear and easy to work with. We expanded the interpreter to include functions to which Wimble made reference but did not flowchart – *inner product, outer product, catenate,* and *index-of.* We made the interpreter extremely portable by having the character set machine (as well as keyboard) independent. We accomplished this by having the program read in the installation's character set from a file at the start-up of the program.

Tables

Rather than using Wimble's method of storing tables in arrays (variable table, value table, function table, token table), we took advantage of one of Pascal's data structures, the *linked list.* This offers two big advantages to the design of the interpreter:

- Array sizes do not have to be declared elsewhere in the program. There is no way of telling which tables will grow very large and which ones will stay small; this is dependent on the calculations being performed with the interpreter and will vary from one terminal session to another. With linked lists, storage allocation is dynamic and can be used for each table as needed (storage is taken from a common pool of storage reserved for linked lists).
- It is a simple procedure to de-allocate storage (using the standard function "dispose" in Pascal) so that it can be re-used by the program as needed. This helps to keep the size of the running program to a minimum.

Values

We store all values as real numbers. We decided to do this based on the fact that although APL's data structures are weak (eg: reals and integers can be stored in the same array), Pascal's data structures are very strongly typed. Numbers are checked to be whole numbers (nonfractional) for certain operations such as index generation (monadic iota) and reshaping (dyadic rho). Numbers are checked to be Boolean for such operations as *logical negation* (tilde), ANDs, and ORs.

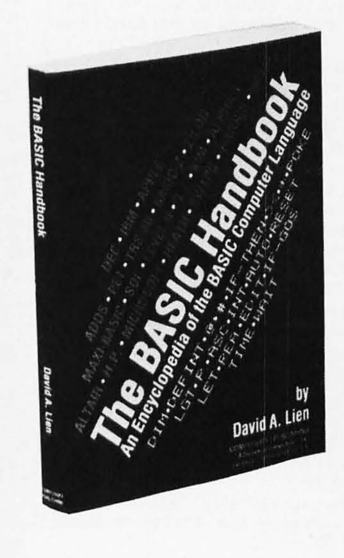
The Nybbles Library is an inexpensive means for BYTE readers to share some interesting but specialized forms of software. These programs are written by readers with small computers and printer facilities, and are therefore designed for particular systems. The algorithms and programming techniques in these programs can be directly used by readers with similar equipment, or can serve as an inspiration for improvisation on computers of different characteristics.

Potential authors of such programs should send us a self-addressed stamped envelope, with a request for a copy of our "Guidelines for Nybbles Authors." Payment for Nybbles items is based on sales and length of the item. Rates are set at the time of acceptance.

Nybbles Library programs are sent in listing form, printed on 8.5 by 11 inch paper on both sides. The Nybbles Library programs are 3 hole punched for collection in loose leaf binders, and come in an attractive folder which serves as a cover.

This month "An APL Interpreter in Pascal" (document # 109) has been added to the Nybbles Library. Use the coupon below to order your personal copy, at $10.00 postpaid in USA or $12.00 for overseas orders.

Book Reviews



The BASIC Handbook: An Encyclopedia of the BASIC Computer Language
by David A Lien
CompuSoft Publishing, San Diego CA, 1978
360 pages, paperback
$14.95

The title of this book really should read *A Dictionary of the BASIC Computer Language.* It is laid out in a self-indexing format as an alphabetic listing of BASIC keywords (such as PRINT, GOTO, and INT) accompanied by a detailed explanation of the effect that the keyword has when used in a program. Operator symbols are also dealt with.

The description given for each keyword includes the following: introductory and descriptive remarks, a test program with a sample run to show how the machine should respond, helpful hints, variations in usage between different brands of computers or different implementations of the BASIC language, and cross-references to related keywords. Also included is a section called "If Your Computer Doesn't Have It." This section is of great value to readers who may have BASIC interpreters that lack certain features.

In many cases this section gives a subroutine which performs a function. These subroutines are similar in design to those which are found in the Radio Shack *User's Manual for Level 1 TRS-80 Microcomputer System.* In some cases a slightly modified algorithm is used for better accuracy. These subroutines are written in a form which transports well between different systems. (The similarity to subroutines in the TRS-80 manual, and a slight emphasis on TRS-80 BASIC, is not surprising. Dr Lien is the author of the *TRS-80 Users Manual.*)

The BASIC Handbook is good within its limits. It will be a help to the beginning programmer, especially one trying to convert a BASIC program from one microcomputer system to another. When this novice programmer encounters a word with which he or she is not familiar in a program, the chances are that it is in this book, along with supplementary information.

A problem arises, however: not all features and differences between BASIC systems occur in the keywords. For example, the BASIC compiler offered by North Star Computers has several characteristics which differ from other BASIC systems. These include reversed use of commas and semicolons, and accessing of single characters from a string by subscript notation. The book could address the punctuation symbol usage, but it does not. The format does not provide a good section to discuss the subscript notation for strings or other differences of a similar nature.

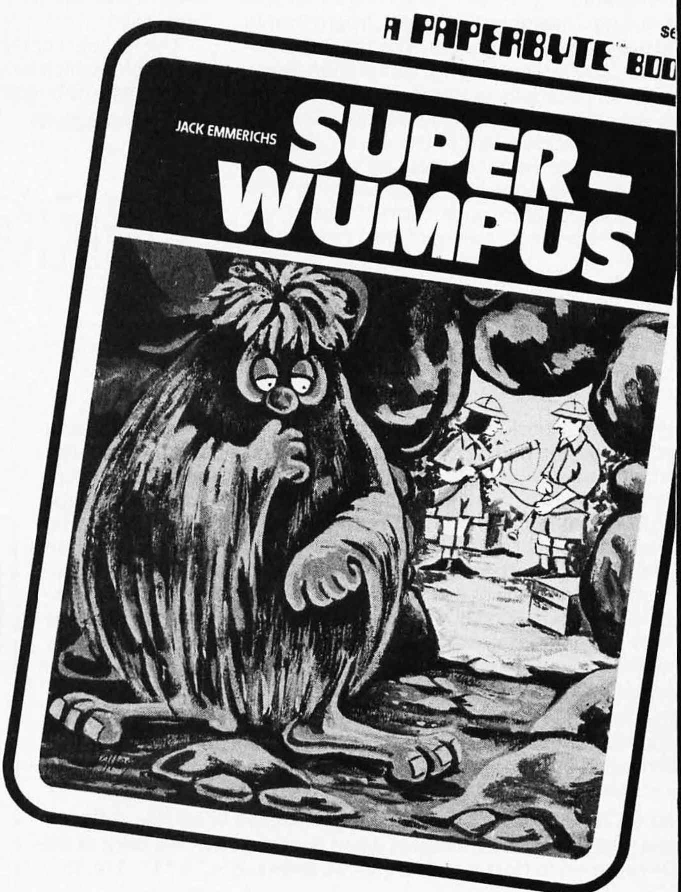
The book in this edition is incomplete. Certain keywords do not appear. Notable by their absence are the string usage statements CHANGE and LINPUT; the special forms RESTORE$ and RESTORE#; the matrix arithmetic operations (MAT C = A + B); the matrix initialization keywords (MAT C = ZER or CON or IDN); the matrix manipulation statements (MAT C = TRN(A) or INV(A), etc.); and most of the various statements for handling data files on mass storage devices.

Part of the reason for the above mentioned omissions is that most of the information presented in this book concerns microcomputer BASIC systems. In particular, the various Microsoft (MITS, Apple, Radio Shack, Commodore, Ohio Scientific) interpreters are well covered. Implementations of BASIC on minicomputers and large mainframes are somewhat neglected, however. They appear in the list on the inside back cover, but most of the more unusual features, those which are most likely to cause trouble, have not been included in this book.

The result of all this is that a person who wishes to convert a BASIC program from a large computer system to a small computer system will need to determine if the more specialized features of the "large" BASIC have been used. If they have, the programmer will need to consult the user's manual for the BASIC language as it works on the large source computer. Dr Lien recognizes this, as he states in the introduction, "*The BASIC Handbook* is not a substitute for the manufacturer's manual which accompanies each computer. It is a supplement."

««« I SMELL A WUMPUS!!! »»»

...SHOOT or MOVE?



ISBN 0-931718-03-1

$6.00

Buy this book at your favorite computer bookstore or order direct from BYTE BOOKS.

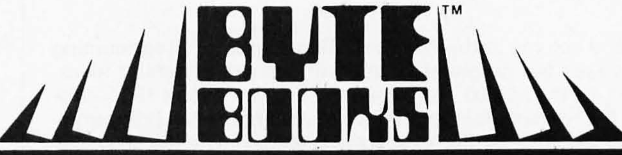
Send 60¢ per book for postage & handling.

It's Jack Emmerichs' new book, 'Superwumpus', an exciting adaptation of the 'Wumpus' game for computer programmers. The cave system of the Wumpus is a dodecahedron, and you're the lost hunter who has to capture the Wumpus to lead you out. You'll have to hurry before your supplies are depleted, or before you're caught in a bat migration, a rock slide, or some other catastrophe.

Superwumpus is outstanding in the realm of computer games as being both challenging and enjoyable.

Superwumpus is programmed in both 6800 assembly language and Basic, and comes complete with source listings and bar codes.

Watch out for Superwumpus—It's addictively fun!



"BOOKS OF INTEREST TO COMPUTER PEOPLE"

70 Main Street, Peterborough, New Hampshire 03458

Circle 36 on inquiry card.

FREE SOFTWARE!

FINALLY....



a quality assembler for the TRS-80*

PRACTICAL MICROCOMPUTER PROGRAMMING: THE Z80
by W.J. Weller

Here from W.J. Weller and Northern Technology Books is the third in the **Practical Microcomputer Programming** series. It is a comprehensive text covering assembly language programming for Z80 based microcomputers. The first 16 chapters cover Z80 programming comprehensively, from binary operations to interrupt handling. Included are chapters on moving data, logical and arithmetic operations, use of the stack, communications with the terminal, floating point arithmetic and graphic output. All programming techniques are illustrated with formal tested examples. An important feature of the book is that it uses the universal standard 8080 mnemonics. This is of great help to users who are upgrading their machines and software to utilize the Z80 processor.

The last part of the book is software; an editor/assembler which will run on any 8080 or Z80 machine and a debugging monitor.

Hardcover $29.95

FREE:

Send in the coupon supplied with the book and receive **FREE** the object programs of the editor/assembler and debug on either **paper tape** or on **cassette tape for the TRS-80 microcomputer.** (Loads in Level I with 16K RAM or Level II.)

Both the book **and** software for only $29.95!

- Also available is the editor/assembler object code for the Z-80 on diskette for North Star disc systems. Only $14.00
- In addition, the editor/assembler and debug source code for the Z 80 are available on diskette for North Star disc systems. Only $14.00
- Or both for $26.00!

AN EDITOR/ASSEMBLER SYSTEM FOR 8080/8085 BASED COMPUTERS
by W.J. Weller and W.T. Powers

This 148 page book contains complete information for initializing and using a powerful new editor/assembler and debugging monitor system, and the full SOURCE text of both. The assembler fully supports all Intel instruction mnemonics as well as the entire language used in **Practical Microcomputer Programming™:The Intel 8080**. The editor/assembler is resident in less than 8K RAM and will run on any 8080, 8085 or Z80 based computer with peripherals which transfer on a character basis or can be made to do so by buffering. The user supplies his or her own I/O drivers. The text editor is extremely simple to use and does not require irrelevant line numbers. Also included is a program to convert Processor Technology™ format tapes to a format usable by the editor/assembler.

This system is not the usual "quickie" software, riddled with errors and limitations, but a professionally created, thoroughly tested and debugged system. At $14.95 it is the best software bargain you are ever likely to see.

AND BY THE WAY...paper tape object copies of all this software are sent FREE to book purchasers when the coupon at the back of the book is returned to Northern Technology Books. 8½" x 11". $14.95

- 8080/8085 editor/assembler object code on diskette for North Star disc systems. Only $14.00
- 8080/8085 editor/assembler and debug source code for North Star disc systems. Only $14.00
- Or both for $26.00!

PRACTICAL MICROCOMPUTER PROGRAMMING: THE INTEL 8080
by W.J. Weller, A.V. Shatzel and H.Y. Nice

Here is a comprehensive source of programming information for the present or prospective user of the 8080 microcomputer, including moving data, binary arithmetic operations, multiplication and division, use of the stack pointer, subroutines, arrays and tables, conversions, decimal arithmetic, various I/O options, real time clocks and interrupt driven processes, and debugging techniques.

This 306-page hardcover book is well worth its $21.95 price and should be in every 8080 or Z80 user's library.

PRACTICAL MICROCOMPUTER PROGRAMMING: THE M6800
by W.J. Weller

This second volume of the **Practical Microcomputer Programming** series addresses the problems of applications programming at assembly level for the M6800. In 16 chapters and more than 100 formal examples , the fundamental techniques of assembly level programming are applied to the solution of specific problems with the 6800. Nowhere theoretical, it is a thorough and detailed methods text for the beginning and intermediate application programmer using the 6800. $21.95 hardcover.

***A trademark of Tandy Corporation**

BITS T.M. inc

Books to erase the impossible

25 Route 101 West,
PO Box 428,
Peterborough, NH 03458
603-924-3355

Please send me the items checked below.

☐ **Practical Microcomputer Programming — The Z80** (with choice of either FREE paper tape or FREE TRS-80 cassette when I mail coupon supplied with book) $29.95
☐ **Editor/Assembler System for 8080/8085 Based Computers** (with paper tape object copies of the software sent FREE when I mail coupon supplied with book) $14.95
☐ **Practical Microcomputing Programming — The Intel 8080** $21.95
☐ **Practical Microcomputing Programming — The M6800** $21.95
☐ Editor/assembler object code for the Z 80 on diskette for North Star disc systems $14.00
☐ Editor/assembler and debug source code for the Z 80 on diskette for North Star systems $14.00
☐ **SAVE $2.00!** Both object code and source code for the Z 80 $26.00
☐ Editor/assembler object code for 8080/8085 on diskette for North Star disc systems $14.00
☐ Editor/assembler and debug source code for 8080/8085 on diskette for North Star disc systems $14.00
☐ **SAVE $2.00!** Both object code and source code for 8080/8085 $26.

Total enclosed $__________(Please enclose 75¢ shipping/handling for each item ordered)

☐ Check Charge Card ☐ Master Charge ☐ Visa

master charge THE INTERBANK CARD   VISA

Card #____________________________
Expires____________ Signature____________________

Name________________________________

Address______________________________

City____________State________Zip____________

Dial your charge card orders toll-free: 800-258-5477 Mon.-Fri 9 - 5 PM (In New Hampshire dial 924-3355)

You may photocopy this page   Dealer inquiries invited

Dr Lien treats the END statement in a way I dislike. He describes it only as a means of terminating program execution; whereas many computers use END as a means of indicating the end of the program text, and some systems use END as a marker for the physical end of file when a program is stored on a disk. Programmers treating END only as an execution terminator scatter ENDs throughout the program. If an unsuspecting user types in such a program on a system using END for end of file and saves it on a disk, he may lose the result of hours of work. I prefer the use of the STOP statement for terminating program execution other than at the end of the program.

I hope that BASIC experts will communicate with Dr Lien, to provide him with exact information concerning the more exotic features of the language. Then, perhaps, the second edition of this essentially helpful book can be more helpfully essential. All things considered, the book is a useful purchase, especially for the beginner, but I urge that it be improved. An improved version could truly require the appellation "encyclopedia."

Richard S Shuford
Editor■

Structured Programming and Problem-Solving with Pascal
by Richard B Kieburtz
Prentice-Hall, Englewood Cliffs NJ 1978
365 pages paperback
$10.95

Good habits appear to be in vogue for 1979. In programming, the good habits that we are suddenly hearing about are documentation, top-down design and bottom-up coding techniques, and the Pascal language. Richard B Kieburtz's book demonstrates the necessity of a firm grounding in the design and implementation of programs in order to cope with the complexity of today's programming problems.

The book is divided as follows: 45 percent Pascal, 45 percent structured design and programming techniques, and 10 percent theory (introductory material on computers that qualifies the book for use as a college textbook).

Pascal is largely defined and taught by the context of its use in solving problems such as determining the intersection of two line segments, writing a word processing program, and running a rabbit population simulation. Although there is an index of

Pascal terms pointing back to the text, many of these terms are not defined in sufficient detail to help debug a Pascal program. The book is readily understandable to anyone unacquainted with the language, but it should not be used as the sole reference book on Pascal as it describes a generalized Pascal that manipulates both character and numeric variables. Specifically, it does not mention some of the more advanced UCSD (University of California at San Diego) extensions to Pascal.

The concept that is the cornerstone for both the theory and implementation of structured techniques in this book is known as "design by stepwise refinement." This automatically implies two techniques that I find best for problems of any complexity: top-down design and bottom-up coding. Top-down design (breaking a problem into manageable subproblems) produces a modular program that can be easily modified. Bottom-up coding (writing the code for every subproblem before writing the code that uses them) avoids the problem of having to rewrite the high level routines to add something that you found you needed at a later date. Kieburtz uses a relatively new flowchart-like notation that depicts the fundamental structured programming constructs (*do-while, repeat-until,* sequence, *if-then-else,* and *case)* in a way that is both graphic and intuitively understandable. For example, the body of a *do-while* clause is a rectangle bordered on the left and top by an L-shaped piece that describes the *while* condition for repeating the block.

The book also introduces several of the better known algorithms and ideas in computer science: the linear interpolation and binary search methods of extracting roots, Gaussian elimination to solve simultaneous equations, backtracking trial and error methods (to solve the eight queens problem), and several simulation examples. The final chapter, "How Does the Computer Work?," deals with binary numbers, machine language, and computer architecture. It is obviously there to catch a larger slice of the textbook market.

All in all, this book is reasonably priced and well worth the money. It is a good introduction to Pascal (but only that), and it exposes the reader to good programming habits on all levels. I wish that I had been exposed to this kind of book when I was learning to program.

Gregg Williams
1605 Eastmoreland #3
Memphis TN 38104■

Revolution in Miniature
by Ernest Braun and Stuart MacDonald
Cambridge University Press, 1978
231 pages hardcover
$16.95

The invention of the transistor was not a fortuitous accident of pure research. It had been sought for years before its actual invention. A working (though impractical) solid state amplifier had been demonstrated as early as 1933, and in 1939 Dr William Shockley tried to put a "grid" in a copper oxide rectifier. It didn't work. When developments in physics finally permitted its invention, the transistor principle was identified within weeks by Bell Laboratories, Purdue University, and a French team, all working independently.

These are some of the fascinating, amusing, and always factual incidents related in *Revolution in Miniature*. The book traces the history of solid-state electronics from the coherer (the first solid-state electronic device) to large scale integration. One of the authors is a historian, the other is a physicist, and I can't think of a better combination for this endeavor. The flair that these two British authors exhibit with their command of the English language provides a force of expression seldom seen in a technical book.

As the publisher states, "Semiconductor electronics' . . . effect on life in the second half of the twentieth century can hardly be overestimated." Solid-state technology has made possible things that were never before envisioned, yet in the beginning the transistor was seen, even by its developers, as a mere substitute for the triode tube (or "valve," as described herein). The later, more successful transistor types were often electrically inferior to the fragile, lab assembled models and were adopted only in the interests of cheap, uniform mass-production. Early integrated circuits contained a lot of hand labor, and digital electronics as we know it today resulted from attempts to minimize the passive components needed in earlier analog circuitry.

The book is heavily footnoted and the bibliography is impressive. Aside from entertainment value, the historical data is well-worth having for reference. You will find answers to such questions as: who developed which manufacturing techniques; which key people spun a new company off from an established one; why Silicon Valley is what it is; and which bar is the scene of employee and information swapping. It's a lot like reading someone's diary.

Revolution in Miniature is a must as a historical reference, and great reading for both electronics types and those who don't care about how things work. It provides first class nostalgia for the old-timers who actually remember using crystal sets, who remember when tubes "went miniature," and who have tried to make equipment smaller by simply cramming conventional parts closer together. As the authors state, at one time it was theoretically possible to achieve a parts density of 1000 per cubic foot, but in practice the heat wouldn't permit it. Our thanks are due to all of those pioneers who made this remarkable revolution possible.

Ernie Brooner
COMLABS
POB 236
Lakeside MT 59922

How to Program Microcomputers
by William Barden Jr
Howard W Sams and Co, Indianapolis 1977
256 pages paperback
$8.95

How to Program Microcomputers is an introduction to machine language programming for the 8080, the 6800, and the 6502 microprocessors. The only other book I have seen that attempted to teach programming for a group of microcomputers did so by presenting all problems in a superset of the PL/M language. This book takes a different approach, and uses the assembler mnemonics for each of the processors discussed.

The book begins with an introduction to microcomputers: what they are, how they operate; and an introduction to alternative processor architectures. The structures of the 8080, the 6800, and the 6502 are described, then alternatives for addressing, memory access, stack manipulation, I/O (input/output) operation and interrupt processing are introduced. Each topic is illustrated with features from the applicable microprocessor.

The next part of the book deals with programming techniques. Data movement, arithmetic operations, multiple precision arithmetic, branching, indexing, subroutines, stack operations, table operations, list processing, bit manipulation, decimal and floating point arithmetic, and I/O are discussed, as well as how to put all of these elements together. As before, examples are given for each processor.

The last part of the book provides standard algorithms for each machine. Twenty different building blocks (most of which belong in any good monitor) are given for each processor. Appendices summarize the instruction sets.

There are three groups of people who might be interested in this book. The first group is composed of anyone becoming acquainted with microcomputer technology who wants an overview of the main processors in present hobbyist use. The second group is composed of people who already have a machine and who want good standard routines. The third group (in which I am included) consists of those hobbyists who have a machine and who would like to see how other processors operate. While I would certainly never trade my Z-80 for any of the processors illustrated in the book, it is good to know how the rest of the world operates.

John A Lehman
716 Hutchins #2
Ann Arbor MI 48103■

Programming Quickies

BASIC Factorials

Alan R Miller
New Mexico Tech
Socorro NM 87801

```
10 REM TEST FACTORIAL SUBROUTINE
20 INPUT "FACTORIAL OF ";X:GOSUB 5000
30 PRINT "THE FACTORIAL OF ";X;" IS ";G
40 GOTO 20
5000 REM GAMMA FUNCTION G(X)
5010 Y0=X+1:Y= Y0+5: Y2= Y*Y*30: Y1=SQR(2*3.14159/Y*Y↑Y
5020 G= Y1*EXP((1-1/(30*Y2))/(12*Y)-Y)/Y0
5030 FOR I=1 TO 4: G= G/ (Y0+I): NEXT: RETURN
OK
```

Listing 1: BASIC program for determining factorials using the gamma function.

```
RUN
FACTORIAL OF ? 2
THE FACTORIAL OF 2 IS 2.00001
FACTORIAL OF ? 3
THE FACTORIAL OF 3 IS 6.00003
FACTORIAL OF ? 4
THE FACTORIAL OF 4 IS 24.0001
FACTORIAL OF ? 5
THE FACTORIAL OF 5 IS 120.001
FACTORIAL OF ? 8
THE FACTORIAL OF 8 IS 40320.1
FACTORIAL OF ? 12
THE FACTORIAL OF 12 IS 4.79001E+08
FACTORIAL OF ? 16
THE FACTORIAL OF 16 IS 2.09228E+13
FACTORIAL OF ? 20
THE FACTORIAL OF 20 IS 2.43292E+18
```

Listing 2: Sample run of the factorial program. Notice that the answers are not exact. The truncated integer portions of the smaller results are exact factorials; as the factorials grow in size, the result quickly exceeds the precision of the floating point representation of the numbers.

Here's another function to add to your BASIC, a factorial calculator. The factorial of a number X is equal to X times X-1 times X-2 etc down to one and is represented by X! Thus 4! is 24. For large values of X, Stirling's approximation can be used to find the gamma function which is readily converted to the factorial by the relation:

$$X! = \Gamma(X+1)$$

To find the factorial of X with the BASIC program shown in listing 1, execute a jump to subroutine at line 5000. On return, the factorial of X will be in G. If for some reason the gamma function itself is wanted, remove the first statement from line 5010 and GOSUB 5000 with the argument in Y0.

The subroutine works by finding the gamma function of a number six values larger than the argument:

$$X! = \Gamma(X+1)$$

$$(X+5) = \Gamma(N)$$

$$= \sqrt{2\pi/N}\ N^N$$

$$\exp\left(\frac{1}{12X}\left(1-\frac{1}{30X^2}\right)-X\right)$$

$$\Gamma(X) = \Gamma(N)/(X(X+1)(X+2)$$

$$(X+3)(X+4))$$

This function is only approximate, as can be seen in the sample run of listing 2. The returned value should be rounded to the nearest integer.■

An 8080

Free Memory Search

William M Hand
18660 Arden Av
Brookfield WI 53005

Since my computer system is continually in a state of flux, I sometimes lose track of the addressing for the various memory boards. To eliminate the hunt-and-seek method of locating unprotected memory blocks, I put together the routine in listing 1 (see page 208) to examine all memory space from hexadecimal 0000 to FFFF and report the start and end addresses of all available spaces.

A memory location exists and is not protected if the processor can write a word to memory and read back the same word. However, since any given memory location may have a value from 0 to FF (the range of the 8080 processor), some care must be exercised in declaring a location as existing and available.

To address this problem, I use a double store routine in which the processor first stores one arbitrary number and then another different number. If the processor reads back the correct number for both stores, that location is a valid unprotected memory cell. [This could also be used as a memory failure check if the two values used were hexadecimal 0000 and FFFF .... RGAC]

Two notes are needed relative to listing 1. First, the line with the pound sign (#) is the link back to the calling routine. If the FMAR routine is called as a subroutine, this line should be replaced with a return instruction. Second, note that upon exiting, the DE register pair points to the next address past the last address pair from the routine. The pointers for start and end of free memory blocks may be pulled out with LHLD or POP instructions.

Also, the routine itself should be located in protected memory (along with the operating system, for instance) since the routine will self-destruct if located in unprotected memory. Be sure to provide sufficient room for the DE register pair to expand.

Total memory requirements for this routine are 66 bytes plus the stack area for the DE register pair storage of free memory boundaries.

If desired, the FMAR routine may be used to simply output the addresses to a Teletype or terminal.■

Listing 1: 8080 assembly language program for finding areas of memory that are free and unprotected. Modifying the command on line 6 allows the program to start looking at any point in memory.

```
0000    *Free Memory Allocation Routine
0001    *       locates available unprotected memory
0002    *
0003    *       coming in: D&E point to location
0004    *       where results are to be stored
0005    *
0006    FMAR:   LXI   H,0      ;SET H&L TO 0000
0007    FMAR1:  CALL  LOOK     ;LOOK FOR FIRST FREE LOCATION
0008            JZ    REP1     ;FOUND IT, REPORT IT
0009    FMAR3:  CALL  CHECK    ;LOOK FOR END OF MEMORY
0010            JNZ   FMAR1    ;NOT YET, LOOK SOME MORE
0011          # JMP   EXEC     ;END OF MEMORY, RETURN TO EXECUTIVE
0012    REP1:   CALL  REPOR    ;REPORT FIRST FREE LOCATION
0013    FMAR2:  CALL  CHECK    ;SEE IF AT END OF MEMORY
0014            JNZ   TRYON    ;NOT YET, CONTINUE LOOKING
0015    FMAR4:  DCX   H        ;REPORT LAST LOCATION
0016            CALL  REPOR
0017            JMP   FMAR3    ;AND USE CHECK TO EXIT
0018    TRYON:  CALL  LOOK     ;LOOK FOR A FREE LOCATION
0019            JZ    FMAR2    ;VALID LOCATION
0020            JMP   FMAR4    ;NOT VALID, LOOK FOR START OF NEXT FREE MEMORY
0021    *
0022    LOOK:   MOV   C,M      ;SAVE MEMORY IN REGISTER C
0023            MOV   M,H      ;WRITE H INTO MEMORY
0024            MOV   A,M      ;READ MEMORY INTO A
0025            CMP   H        ;SEE IF A AND H AGREE
0026            RNZ            ;NO, NOT VALID MEMORY
0027            MOV   M,L      ;OK PASS 1, THIS TIME PUT IN L
0028            MOV   A,M      ;READ BACK INTO A
0029            MOV   M,C      ;PUT C BACK INTO MEMORY
0030            CMP   L        ;SEE IF A AND L AGREE
0031            RET            ;RETURN WITH FLAGS SET
0032    *
0033    CHECK:  SUB   A        ;SET A TO 0
0034            INX   H        ;INCREMENT H&L
0035            CMP   H        ;SEE IF H=0
0036            RNZ            ;NOT YET, RETURN
0037            CMP   L        ;SEE IF L=0
0038            RET            ;AND RETURN
0039    *
0040    REPOR:  MOV   A,L      ;GET THE LOW ADDRESS
0041            STAX  D        ;STORE WITH D&E POINTER
0042            DCX   D        ;DECREMENT POINTER
0043            MOV   A,H      ;GET HIGH ADDRESS
0044            STAX  D        ;STORE WITH D&E POINTER
0045            DCX   D        ;DECREMENT POINTER
0046            RET            ;RETURN
```

---

5 Byte Hexadecimal to ASCII Converter

Ashwin L Doshi
5830 Green Valley Cir 105
Culver City CA 90230

I was recently challenged by a colleague to find the most efficient 8080 code to translate hexadecimal 0 thru F (stored in the accumulator) into ASCII 0 thru 9 and A thru F (also in the accumulator). After I came up with a 5 byte translation, he showed me a well-published 6 byte translation (of which I was unaware) which is as follows:

```
ADI  90H
DAA
ACI  40H
DAA.
```

The 5 byte code that does the same translation is as follows:

```
DAA
ADI  F0H
ACI  40H.
```

The latter assumes that the carry and the auxiliary carry are reset, which is the case in all applications that I could find of this translation.■

Checkbook Balancing Routine

Loring C White
26 Boswell Rd
Reading MA 02119

Every month the bank statement arrives and we have to go through cancelled checks and the usual mathematical ritual to reconcile our figures with those of the bank. Here is some software the computer enthusiast can use to balance a checkbook. The program in listing 1 is written in MITS 8 K BASIC Revision 3.2 (used on my Altair 8800 computer). *[Since the MITS 8 K BASIC language was written and implemented by the Microsoft Company, this same listing should work with minor changes on a number of computers besides the Altair. These include the Apple II with the "Applesoft" BASIC, the Radio Shack TRS-80 with Level II BASIC, and the Commodore PET computer. . . .CH]* If you have printer or Teletype, you can get hard copy of all pertinent information for later references.

The program has the following features:

- The initial printout is a listing of all outstanding checks by check number, date and amount.
- A list of all cancelled checks as they are entered as well as a final summary list is given.
- A new, updated list of outstanding checks is provided to update the list of checks appearing in the data statements. Provision is made for this listing to be in data format so that it can be punched on tape to make the program update easier.
- The computer will search for each check listing as it is entered during the program run.
- Input statements are provided for entering the bank statement balance; service charge and deposits not entered on the statement.

```
3   DIM P(100)
4   DIMN(100),D(100),A(100),NR(100),DR(100),AR(100)
5   PRINT"PROGRAM TO BALANCE CHECK BOOK"
7   PRINT"ONLY CHECKS NOT RETURNED SHOULD BE LISTED IN DATA 600"
8   FORX=0TO70:PRINT"-"; :NEXT:PRINT
9   PRINT"CHECK#","DATE(MDY)","AMOUNT"
10  FORN=1TO100
20  READ N(N),D(N),A(N)
21  IFN(N)=0THEN50
25  S=S+A(N)
26  PRINT N(N),D(N),"$";A(N)
40  NEXT
50  PRINT"LIST ALL CHECKS RETURNED: CHECK#,DATE(MDY),AMOUNT($)"
55  PRINT"LIST '0,0,0' AS LAST CHECK IN LIST"
60  FORN=1TO100
61  INPUT NR(N),DR(N),AR(N)
65  FOR X=1TO100
67  IF NR(N)=0THEN130
70  IF NR(N)=N(X)THEN80
75  GOTO95
80  IF DR(N)=D(X)THEN90
85  GOTO95
90  IF AR(N)=A(X)THEN100
95  NEXT X
97  NEXT N
100 PRINT"CANCEL CHECK#:";NR(N);"DATE:";"AMOUNT:$";AR(N)
102 P(Y)=N
103 Y=Y+1
110 T=T+AR(N)
120 GOTO97
130 PRINT"TOTAL AMOUNT IN $ OF CHECKS RETURNED FROM BANK=$";T
134 PRINT"LIST OF CANCELLED CHECKS"
135 PRINT "CHECK#","DATE(MDY)","AMOUNT $"
136 FORY=0TO100:IFN(P(Y))=0THEN140
137 PRINT NR(P(Y)),DR(P(Y)),AR(P(Y))
139 NEXT
140 PRINT"TOTAL CHECKS NOT RETURNED=$";S-T
150 INPUT"ENTER BALANCE PER STATEMENT FROM BANK $";B
160 INPUT"TOTAL OF DEPOSITS NOT CREDITED ON STATEMENT $";D
166 INPUT"ENTER SERVICE CHARGE INDICATED ON BANK STATEMENT $";SC
168 Z=B-S+T+D+SC
170 PRINT"CHECKBOOK BALANCE SHOULD BE=$";INT(Z*10↑2+.5)/10↑2
180 PRINT"REM TO DELETE ALL RETURNED CHECKS FROM DATA LIST"
190 PRINT"REM TO SUBTRACT SERVICE CHARGE FROM CHECKBOOK BALANCE"
191 PRINT"IF YOU WANT LIST OF CHECKS OUTSTANDING FOR NEW DATA"
192 PRINT"LISTING THEN PREPARE TELETYPE TAPE LEADER AND TYPE 'YES' ";
193 INPUT  V$:IF V$="YES"THEN200
194 GOTO500
200 FORN=1TO100
210 FORY=0TO100
220 IFN(N)=0THEN500
230 IFN(N)=NR(P(Y))THEN250
240 GOTO280
250 IFD(N)=DR(P(Y))THEN270
260 GOTO280
270 IFA(N)=AR(P(Y))THEN290
280 NEXTY
285 GOTO300
290 NEXTN
300 PRINT600+L;"DATA";N(N);",";D(N);",";A(N)
310 L=L+1
320 GOTO290
500 END
600 DATA 100,12876,18.75
601 DATA 3,3177,2.6
602 DATA 6,3177,16.2
603 DATA 7,3177,48
604 DATA 8,3177,16.75
605 DATA 10,3177,251
606 DATA 13,32177,70
607 DATA 14,32877,70
608 DATA 15,31477,70
609 DATA 16,31577,15
610 DATA 17,3777,12
611 DATA 18,3977,5
612 DATA 19,3977,5
613 DATA 100,31077,88.4
614 DATA 100,31177,15.62
615 DATA 20,31177,20
616 DATA 100,31077,8
617 DATA 21,31277,47
618 DATA 1,31277,52
619 DATA 2,31277,150
620 DATA 100,31477,9.93
700 DATA 0,0,0
```

Listing 1: Checkbook balancing routines for Microsoft (MITS) BASIC. The data statements contain all outstanding checks that will be checked.

- A final summary is provided giving the total of all outstanding checks, checkbook balance and the checks returned from the bank.

How It Works

To implement the program it is necessary to provide a list of all checks written by number, date, and amount in the data statements at the end of the program. When I first started writing the software I included the name of the company but later discovered that this information is not really needed.

The first data statement in the program of listing 1 is:

```
600 DATA 100, 12876, 18.75
```

The statement says that check number 100 was written on December 8, 1976 for the amount of $18.75. (I usually carry a number of blank checks in case I need to write a check. I always number this type of check with 100. At the end of the month I may have several checks with number 100 but this is no problem, because they are also identified with the date and amount.) The computer, when searching for each check, looks for all three pieces of information before assuming that the check has been located. Listing each check on a separate line uses up more memory, but there are good reasons for doing this that become apparent when you run the program.

Using the Program

After all the checks are listed in the data statements, as shown in the program, you are ready to run the program. Take the cancelled checks you get from the bank and input the check number, date and amount the same way you entered the information in the data statements. Then hit carriage return. The program will search the data list for the check and deduct it from the balance, printing for example:

```
"CANCEL CHECK #: 100
DATE: 12876
AMOUNT: $18.75"
```

If the check is not located in the data, the computer types a question mark meaning that either the check isn't there or you have not entered the data correctly.

After all the checks have been entered, type 0,0,0 for the last check and hit carriage return. The computer will give you a complete list of all the cancelled checks just entered plus the total of all the outstanding checks. You will be asked to enter the bank statement balance and any deposits not shown and the service charge, if any. Using this information, the program calculates the balance in your checkbook. In this way you can reconcile your arithmetic with that of the bank.

Normally at this point it is necessary to change the data statement list by eliminating all the cancelled checks received from the bank. This would mean searching and typing some of the line numbers. When I developed the program I decided to let the program do this work, so you will be asked if you want an updated "data" list of the outstanding checks.

If you have a mass storage device, you can store the data statements. The program lists all the outstanding checks in the required data format, including new line numbers. The program can then be updated by entering the information back into the program. All data statement numbers not stored will have to be deleted by hand. This is now an easier job because these numbers are at the end of the program and no searching by the operator is required. Also, don't forget to deduct the service charges from your checkbook balance.

Before developing this software I used to dread receiving "that envelope" from the bank, but now I actually look forward to it in spite of the fact that it requires a bit of effort to enter the required information into the computer.■

A Peek at Poke

M Parris
Dept of Chemistry
U of Victoria
POB 1700
Victoria BC
CANADA V8W 2Y2

Users of TRS-80 Level II BASIC will find the POKE function handy for the occasional manual patching of object programs. However, it's frustratingly slow and annoying to be forced into decimal notation. Listing 1 is a Level II BASIC program residing in upper limit statement numbers for a 16 K byte

```
65509   CLS:PRINTCHR$(14):L%=32767
65510   GOSUB65512
65511   POKEL%,D%:L%=L%+1:GOTO65510
65512   GOSUB65528
65513   IFA$=":"THENI%=1ELSEI%=0:IFI%=0GOTO65522
65514   GOSUB65528
65515   GOSUB65529
65516   D%=A%*4096
65517   GOSUB65528
65518   GOSUB65529
65519   D%=D%+A%*256
65520   GOSUB65528
65521   GOTO65523
65522   D%=0
65523   GOSUB65529
65524   D%=D%+A%*16
65525   GOSUB65528
65526   GOSUB65529
65527   D%=D%+A%:IFI%=0 THENRETURNELSEL%=D%:GOTO65512
65528   A$=INKEY$:IFA$=""THEN65528ELSEPRINTA$;:IFA$=
        " "THEN65528ELSEA%=ASC(A$):IFA%=10THEN65528E
        LSEIFA%=13THEN65528ELSERETURN
65529   A%=A%-48:IFA%<0 THENSTOPELSEIFA%<10THENRETURN
        ELSEA%=A%-7:IFA%<16THENRETURNELSESTOP
```

Listing 1: Level II BASIC program for the TRS-80 which allows hexadecimal data to be loaded into memory.

```
10    X=32+RND(159):REM GET A RANDOM NO. BETWEEN 32 AND 191
20    FORJ=0TO1023:POKE15360+J,X:NEXTJ:REM FILL THE SCREEN
30    FORI=1TO1000:NEXTI:GOTO10:REM WAIT AWHILE AND DO IT AGAIN
```

Listing 2: Example of a program to fill the screen of the TRS-80 with graphic characters.

```
ORG 16526D                   ;LOCATION OF USER FUNCTION ADDRESS
            DW START         ;USER FUNCTION ADDRESS
ORG 32000D                   ;THIS LOOKS LIKE A NICE PLACE
            DS 1             ;DATA TO BE TRANSFERRED TO USER FUNCTION
START:      LXI H,32000D     ;GET THE DATA TO BE TRANSFERRED -
            MOV C,M          ;INTO C
            LXI H,3C00H      ;VIDEO DISPLAY STARTS HERE (HEX)
            LXI D,0400H      ;SIZE OF DISPLAY (HEX)
NXT:        MOV M,C          ;LOAD THE DATA FOR DISPLAY
            DCR E            ;COUNT THE -
            JNZ EOK          ;REMAINING -
            DCR D            ;LOCATIONS IN -
            RZ               ;VIDEO DISPLAY -
EOK:        INX H            ;AND FILL THEM ALL -
            JMP NXT          ;WITH THE DATA
END
```

Listing 3: An assembly language program which can be called by the modified version of listing 2 to fill the screen quickly. Use the hexadecimal loader to put the object code of this program into memory.

TRS-80 system, which enables fast keyboard entry of hexadecimal code.

The starting address is entered as :HHLL followed by the successive data bytes, which may be spaced as appropriate for clarity in on-screen checking. An invalid character causes a return to the COMMAND mode. The following trivial example illustrates its use.

Example

The BASIC interpreter is too slow for dynamic graphics, as the program in listing 2 illustrates. Run it and see how slowly the screen loads. Now replace statement 20 by:

```
20 POKE32000, X:X=USR(0):
REM FILL THE SCREEN FASTER,
```

which calls the machine language subroutine in listing 3 to do the same job much faster. Insert the assembled object code using the hexadecimal loader, noting that memory size must now be less than 32000:

```
: 408E 017D
: 7D01 21007D4E21003C110004711DC2127D15
       C823C30B7D
```

Now run the program again.■

Clubs and Newsletters

Attention: Buffalo NY Apple II Owners

Gary Weir has written from Buffalo NY with information about a new Apple II users group in his area. Called the Apple Byters Corp, they are highly motivated to provide new owners of the Apple II with the help they may need to successfully program and utilize their investment. A booklet is currently being prepared combining insights and solutions to bugs previously encountered by the club's members. A copy and its updates are included in the $5 membership fee. Apple II owners or potential owners should contact Gary at 225 Walton Dr, Snyder NY 14226 concerning meeting times and locations.

Apple Bay Area Computer Users Society

ABACUS (Apple Bay Area Computer Users Society) meets the second Monday of each month at the Hayward BYTE Shop, 1122 B St, Hayward CA. They have an active membership of 40 and have developed a club library of 200 programs. They are negotiating to trade libraries with several other clubs. Membership is $12 a year which includes a monthly newsletter. Contact Ed Avelar, president, at (415) 583-2431 or David Wilkerson, secretary, at (415) 482-4175.

Mexican Computer Club

We have been notified of the existence of a Mexican computer club. Called the Microcomputer Club, this group is primarily concerned with the Apple II and OSI products. They are interested in exchanging information and experiences with other computer groups. Contact Alfredo Buzali, fte de Quijote #5, Mexico 10, D F or call 5-89-22-79 between 7 and 8 PM.

Central Alabama TRS-80 Computer Society

Several TRS-80 users in Montgomery AL have formed the Central Alabama TRS-80 Computer Society. They are planning a club library, a local newsletter and a club computer. Another aim is to provide each new member with a membership package which would contain magazine subscription blanks, addresses of hardware and software suppliers, and other information pertaining to the TRS-

80. The meetings are held on the third Tuesday of each month at various locations around Montgomery. Contact Walter F Bray, 2073 Rexford Rd, Montgomery AL 36116.

Newsletter for Sorcerer Owners

Orders are now being accepted for an independent user newsletter dedicated solely to the Exidy Sorcerer. The $15 subscription price includes all ten issues of volume one, and the first issue will be available around July 1st. *The Source* will include items of general interest to Sorcerer owners, such as program listings, how-to-do-it articles, and hardware and software reviews. Contact ARESCO, POB 1142, Columbia MD 21044.

Sorcerer Users Group

Computer Mart of Massachusetts has announced the formation of the Sorcerer Users Group. The purpose of the group is to set up a channel of communication between Sorcerer owners and to provide information on hardware and software developments to the Sorcerer user. The group has a membership of about 30 people. The $5 membership fee includes the monthly newsletter, *The Exidy Monitor.* Contact Computer Mart of Massachusetts Inc, 1395 Main St, Waltham MA 02154.

North American Computer Association

The North American Computer Association (NACA) recently began its third year with a membership of approximately 20 independent businessmen in the computer systems field. One of the objectives of the organization is to increase the efficiency of each member's individual organization by pooling all the different programming developments and the selling and servicing techniques used by the various members. NACA meets once a month in Dallas TX, and interested businessmen are welcome to attend. Contact Tom Crites, Suite 811, 1001 Main St, Lubbock TX 79401.

Small Computer Users Join England's Central Program Exchange

The Central Program Exchange (CPE) at The Polytechnic, Wolverhampton, is opening its doors to users of small computers in an effort to coordinate the free interchange of programming. The Exchange has 72 members, and currently holds a library of over 200 programs in BASIC, FORTRAN and ALGOL. Individual members can obtain CPE services for $10 per year. This entitles members to a catalogue, newsletter and up to ten CPE programs as listings or paper tapes. As the personal element expands, CPE also proposes to offer cassette versions at a small extra charge. Contact Dr G Beech, Central Program Exchange, Dept of Computing and Mathematical Sciences, The Polytechnic, Wolverhampton, ENGLAND WV1 1LY.

Software Magazine for Microcomputers

The *Software Exchange* magazine provides information about the latest software that has been developed. Included are reviews, abstracts and articles about software for today's microcomputers. The Software Exchange is published bimonthly. Subscriptions are $5 per year in the US, Canada and Mexico. International subscriptions are $19. Contact The Software Exchange, POB 55056, Valencia CA 91355.

Free Newsletter Features Computer Product Marketing Information

Crossroads is a free periodic newsletter which highlights tools and techniques in selling software, data services and turnkey systems. *Crossroads* provides helpful hints, insights to trends, results of experiments and generally, the successes and failures associated with development, sale and support of computer based products. Contact Editor, *Crossroads*, Cross Associates, Suite 530, 9000 Keystone Crossing, Indianapolis IN 46240.

6800 Users Newsletter

The Chicago Area 6800 Users Newsletter is a monthly publication aimed at providing information and assistance to those users of the 6800 microprocessor. For further information, contact Phillip Schuman, 1354 Finley, Lombard IL 60148. ■

BYTE's Bugs

A Computer by Any Other Name

A typographical error in "Build a Computer Controlled Security System for Your Home: Part 3" by Steve Ciarcia (March 1979 BYTE, page 150) may have caused some head scratching among our readers. The caption for figure 1 contained the following sentence: "Op amp IC2 is used as a computer to convert the output accordingly." The sentence should have read as follows: "Op amp IC2 is used as a *comparator* to convert the output accordingly." Mr. Ciarcia did not attempt to construct his circuit using a programmable op amp. ■

Circle 320 on inquiry card.

Circle 32 on inquiry card.

Circle 78 on inquiry card.

Languages Forum

Comments on " A High Level Language for 8 Bit Machines "

Glen Newton
Sperry-Univac
Roseville MN 55113

Languages Forum is a feature which is intended as an interactive dialog about the design and implementation of languages for personal computing. Statements and opinions submitted to this forum can be on any subject relevant to its purpose of fostering discussion and communication among BYTE readers on the subject of languages. We ask that all correspondents supply their full names and addresses to be printed with their commentaries.

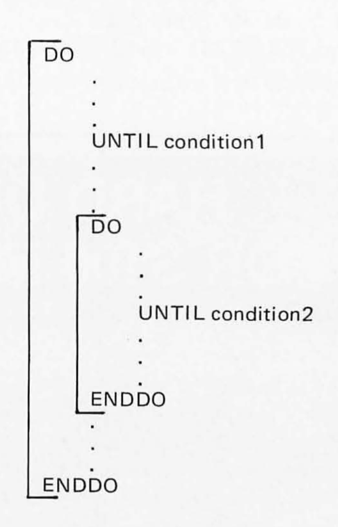
"A High Level Language for 8 Bit Machines," by Ted Williams and Steve Conley (July 1978 BYTE, page 152) discusses the interpreter for a simple language. If we take the language as given and ignore minor errors in the examples and flowcharts (such as pushing a variable "near operand stack" rather than onto it), several substantial errors remain.

First, evaluation is claimed to start "in the innermost parentheses." Following the flowchart in figure 2 or table 2 or the example in listing 3 shows that this is not true, despite the incorrect annotation accompanying listing 3. Expressions are evaluated from left to right until parentheses are encountered. In order for evaluation to begin within the innermost parentheses, in the language described in the article, each binary operator except the last would have to be followed by a left parenthesis. Since the programmer cannot define his own functions, the result of evaluation will be the same as if evaluation had begun in the innermost parentheses *if* the language's intrinsic functions have no side effects.

Second, the claim that although interpretation provides some advantages, "the price paid for this feature is memory" is misleading. The combination of source code,

data, tables, and the interpreter program is often smaller than the compiled code and data, particularly if the source language is a powerful one, like APL, or contains space consuming operations not supported by hardware, such as A↑B, where A and B are floating point numbers.

Third, the language presented in the article allows nested DO loops, but the use of UNTIL in this context can lead to problems since an UNTIL within two or more loops is not syntactically connected with a specific loop. For example, the second UNTIL statement in example 1 could express the programmer's intention to leave either the inner or the outer loop when the condition is satisfied. Defining the UNTIL statement semantics to require exit of the innermost enclosing loop when the condition is satisfied solves this problem but it does not correct the flaw in the DO routine in figure 5 of the article. When an UNTIL condition is satisfied, the interpreter seems to search for the first available ENDDO, rather than the matching one. In general this will not work properly, as illustrated by considering the first UNTIL statement in example 1.



Example 1: Nested DO loops.

Fourth, the authors claim that "the lack of statement labels excludes the possibility of errors caused by not nesting DO loops within each other (which is possible in a language like FORTRAN)." In fact, the possibility exists; just take any appropriate FORTRAN nonnested, overlapping DO loop example and remove the statement numbers to see how the error is possible. The difference is that with statement numbers or labels the compiler or interpreter would have sufficient information to detect the error; without them the error must go undetected because the resulting program is syntactically correct.

Fifth, like the factorial function which is presented to illustrate recursion in programming primers, the use of recursion to find matching ENDIFs during interpretation is unnecessary overkill. The nonrecursive use of an "unmatched IF" counter is adequate and conceptually simple. Furthermore, it avoids the problem that "care must be taken in allocating and preserving local data within SEARCH during recursion."

Finally, although APL, like the language presented in the article, does not use operator precedence within expressions, APL expressions are evaluated from right to left, not left to right. Furthermore, APL has numerous nonstandard operators, providing some justification for its no precedence evaluation; in contrast, the language presented has only standard arithmetic operations plus functions. Thus the claim that the use of precedence in this language would lead to confusion is probably unfounded. Probably the best justification for the decision to have no precedence is that it simplifies implementation somewhat. However, the authors' claim that "this procedure minimizes the size of the stack" is incorrect, as shown by example 2. The FORTRAN expression A − B * C requires parentheses to be expressed in a precedenceless language. The parentheses, in turn, require an extra level on the operator stack (in general, one for each level of nesting). Example 2 shows that a precedenceless language can, for some expressions, require *more*, not less, stack space than a language using precedence.■

Language	Expression	Operator Stack (Maximum)	Operand Stack (Maximum)
FORTRAN	A − B * C	* −	C B A
precedence-less language	A − (B * C)	* (−	C B A

Example 2: Stack size comparison.

Languages Forum

SNOBOL Conquers All?

Bruce Burns
3852 Amundson Av
Bronx NY 10466

In my experience (at the Bronx High School of Science and at home with my own 6502 system) the most enjoyable part of computing is the coding of algorithms. I know several languages, and I find some beauty in almost every language. The languages I know are BASIC, FORTRAN, LISP, APL, PL/I, and SNOBOL4. BASIC has simplicity, FORTRAN has the virtue of speed (FORTRAN compilers have developed quite a bit over the years), LISP has a straightforwardness unparalleled in most other languages; PL/I has the virtue of strong structure, and while I know almost no Pascal, it is obvious that it, too, shares this virtue. APL has sheer array processing power and great elegance with its implicit looping and other simplifying features. And, finally, I get to SNOBOL4, by far my favorite.

I feel that SNOBOL4 is one of the most powerful languages in existence. Like APL, it is loaded with elegance and implicit looping, and, also like APL, it lacks the standard structures for repetition of BASIC, FORTRAN, PL/I, and Pascal such as the FOR-TO or DO loops. These structures are not needed as much in SNOBOL4 programs. In addition, the language is powerful enough to permit user defined functions which are implementations of these structures. An example of the powerful implicit looping coupled with some explicit looping is the following statement, which will (in the full scan mode) permute the characters in the string S so that they are in increasing lexical order. This is useful in certain applications which involve set operations:

```
LEX_ORD S LEN(1) $ A LEN(1) $ B
*LGT (A,B) = B A :S(LEX_ORD)
```

SNOBOL4's major feature is its dexterity with string manipulation, particularly the operation of pattern matching (ie: the language is good at scanning strings; looking for patterns; and processing them when found). But in addition to these capabilities, SNOBOL4 yields considerable power with respect to data manipulation. I know of no other language (not even APL, which specializes in array manipulation) that allows an array to have an integer in one element, a real number in the next, a string in the next, another array in another, *itself* in still another, etc. One may put *any* datatype into *anything*, and one may also create user defined datatypes if desired. The reaction of some people to all this is indifference, but they are missing the point; the best part is knowing you can do it. Besides, one of the most important applications of a language of SNOBOL4's string processing capabilities is implementation of experimental languages, and with these other abilities, one may create all sorts of arbitrarily complex languages.

While all these abilities may seem confusing at this time, when they are fully understood they are simple to use and the good programmer will soon master them. Opponents to the language say they feel that the language's power invites unstruc-

tured programming because facilities such as run time symbol tables and run time SNOBOL4 code compilation are easy powers to get carried away with. But if programmers take the time and care to know the language they are programming in, they will be in full control. It is possible to create well-structured functions which even redefine themselves during execution. This can simplify a program considerably, but it must be done carefully or the program will be an undebuggable mess.

I recently wrote a function in SNOBOL4 that takes the inner product of arrays of arbitrary dimensions—that is, of any size and any number of dimensions. In APL, this function is a primitive (built-in) function, but I challenge anyone to produce the same in BASIC, FORTRAN, or PL/I. Like APL's generalized inner product, my function allows any two operations to be utilized in the formation of the product array. The definition in SNOBOL4 is easily followed. Upon call, the function goes through some initialization, then redefines itself and calls itself recursively for each dimension of the involved arrays. In this manner, it is evident that the function will work as well for multidimensional arrays as for vectors, and verifying it for vectors is very easily done. This function is a good example of the freedom with which SNOBOL4 processes all sorts of datatypes.

I cannot deny that SNOBOL4's powers may be easily abused, so I must warn that it is a language only for programmers who can discipline their own thinking and don't need a computer language to force them to do so. For those people, SNOBOL4 will perform amazingly well.

SNOBOL4 has remained a language found *only* in large batch computers for too long. It is time to introduce it to the personal computer enthusiast, who can use its great powers to his or her own ends. (It has been found that SNOBOL4 is extremely good for game programming.) And so, any fellow BYTE reader who knows and loves SNOBOL4, please join me in my crusade for SNOBOL4 on microcomputers. I am aware of the fact that there are real reasons why SNOBOL4 has thus far run only on large computers, but these troubles must be overcome. I also hope I have sparked interest in any potential SNOBOL4 users reading this. If what I have described intrigues you, look into the language. You won't be sorry.■

Random Comments

David Thornley
2600 Bryant Av S, #205
Minneapolis MN 55408

In the September 1978 BYTE ("Letters," page 17), Scott Johnson inquired about the production of random numbers, particularly hardware-generated ones. The first and foremost consideration in generating and using random numbers is to access Donald E Knuth's *The Art of Computer Programming*: Volume 2, *Seminumerical Algorithms*. Most of what I am going to say is derived from there.

First, beware of assuming that a process, whether hardware or software, produces random numbers simply because the process by which it produces numbers is incompletely understood. If you wish to use such a source, test the results for randomness using every test you can think of. There are many good ones in Knuth's book. The Z-80 refresh register, by the way, can be a good source for one random number, but is not recommended for more than one. As far as I can tell, the refresh counter is incremented every time an instruction fetch occurs, which makes it rather deterministic, although if a variable delay is imposed (such as waiting for an entry from the keyboard) it could be used.

For outside randomness, just look around you. Plug your computer into your stereo receiver, set the selector to AM, find a frequency away from any broadcasts, and fiddle with the volume until you get random bits from the static. (To even out the distribution of low and high bits, take the transitions from low to high and high to low as your bit input — in other words, take 2 bit signals, throw out 00 or 11, and treat 01 as 0 and 10 as 1.) This may or may not work, but if it does work it is a quick way to randomness. Give some thought to encouraging noise in the system. Read random numbers off the cassette of your favorite rock group . . . be creative.

For those, like myself, who know something about programming but are lost with hardware, this formula is taken from Knuth's book cited above:

$$"X_{n+1} = (a \times X_n + c) \bmod M"$$

where M is the word size you are using (probably hexadecimal 10000), a is between hexadecimal 300 and 7C00, and has 5 or D as its last digit, and c is odd and somewhere in the neighborhood of hexadecimal 3800. The calculation must be performed exactly, which is much easier in assembler than in BASIC; in the former, one merely disregards the inevitable overflow. Starting with X_0 at any initial value, this will give a long series of good pseudorandom numbers.

Finally, for people with RND functions, here are several tips: Generating random numbers until the user hits the BREAK key (or otherwise inputs something) is a great way to get a random seed as long as the system does not somehow reinitialize the seed before the program uses the random numbers. Or, to make an intrinsically questionable generator work, throw out a random number of numbers as follows:

```
LET J=INT (10*RND(0)+1)
FOR I=1 TO J
LET X=RND (0)
NEXT I
```

This would fit well in a program as a subroutine, to be called whenever a random number was desired.

I hope this information will be of use to some people.■

Event Queue

June 2, University of Wisconsin — Parkside Computer Fair III, Kenosha WI. Contact Don Piele, UW-P Computer Fair III, University of Wisconsin-Parkside, Kenosha WI 53141.

June 3-6, 1979 International Summer Consumer Electronics Show, McCormick Place, Chicago IL. This show serves as the marketplace for the entire consumer electronics industry. Contact Consumer Electronics Show, 2 Illinois Center, Suite 1607, 233 N Michigan Av, Chicago IL 60601.

June 4-5, Computer Cryptography, Massachusetts Institute of Technology, Cambridge MA. Instruction in the Data Encryption Standard and the new public key cryptographic systems will introduce a working knowledge of the use of cryptography in computer applications. Managers and operators of computer systems will also become acquainted with economic and implementation issues as well as techniques for using this new science in communication networks. Contact MIT, Center for Advanced Engineering Study, Cambridge MA 02139.

June 4-6, Laser Beam Information Systems, New York NY. This seminar will cover the application of laser technology to image and data manipulation in the form of scanning, transmission and reproduction. It will lead the student through the principles and practices of laser beam information systems in preparation for direct application to such fields as facsimile, computer memory and display, target identification, reconnaissance, photo composition, and image manipulation. Contact The University of Chicago, Center For Continuing Education, 1307 E 60th St, Chicago IL 60637.

June 4-7, 1979 National Computer Conference, New York Coliseum, New York NY. NCC '79 will feature a premier showcase of the state of the art in computing and data processing. Leading organizations, large and small, will show the latest equipment and services in approximately 1500 booths. More than 100 program sessions are planned, emphasizing the four major areas of management, applications, science and technology, and social implications. In conjunction with NCC '79, a Personal Computing Festival of commercial exhibits, application demonstrations, and technical sessions on microcomputer systems and applications will be held at the Americana Hotel. Contact NCC '79, c/o American Federation of Information Processing Societies Inc, 210 Summit Av, Montvale NJ 07645.

June 6-8, Twelfth Annual Association of Small College Computer Users in Education Conference, Denison University, Granville OH. Sessions will include the presentation of papers, demonstrations of the educational use of microcomputers, computer text book surveys, discussions with authors of computer texts, administrative uses of computers in small colleges, and a tutorial on microprocessors. Contact Douglas Hughes, Computer Center, Denison University, Granville OH 43055, (614) 587-0810.

June 6-8, Eighth Annual Conference of the MUMPS Users Group, Marriott Hotel, Atlanta GA. Papers will be presented on all aspects of MUMPS development, implementation, and use. Contact Judith Faulkner, Program Committee, Department of Psychiatry, Clinical Sciences Ctr, 600 Highland Av, Madison WI 53792.

June 6-8, Computer Contract Negotiation, New York NY. This three day course is designed to give participants sound answers to the complex ramifications of preparing and negotiating computer contracts. Contact Brandon Consulting Group Inc, 505 Park Av, New York NY 10022.

June 11-13, Minicomputers and Distributed Processing, Chicago IL. This seminar will examine the uses, economics, programming, and implementation of minicomputers. Contact The University of Chicago, Center For Continuing Education, 1307 E 60th St, Chicago IL 60637.

June 13-15, Computer Applications in Industry, Grenoble FRANCE. This symposium is intended as a forum for the discussion of recent advances in the applications of computers to industrial processes. The symposium will cover basic problems in computer science as related to industrial applications. Contact ALPES Congres, Avenue d'Innsbruck, 38029, Grenoble-Cedex, FRANCE.

June 19-21, International Microcomputers/Minicomputers/Microprocessors '79, Palais des Expositions, Geneva SWITZERLAND. Focusing on the changing state of the art in mini/microcomputers and microprocessors, the 1979 conference program will probe advances in systems and equipment with emphasis on practical applications and uses of minicomputers and microcomputers as well as the techniques important to their development.

June 19-23, First Annual Meeting of The American Association of Physics Teachers, New Mexico State University, Las Cruces, NM. The theme of this meeting will be the use of microcomputers in physics teaching. Contact American Association of Physics Teachers, Graduate Physics Building, SUNY at Stony Brook, Stony Brook NY 11794.

June 20-22, The 1979 Symposium of the Wilmington Section of the Instrument Society of America, University of Delaware, Newark DE. The symposium theme: "Measurement Technology for the 80s," is being programmed by three of ISA's divisions: Process Measurement and Control, Analysis Instrumentation, and Water and Waste Water Industries. Contact A H Straightiff, E I DuPont de Nemours and Co Inc, (302) 366-3810.

June 25-26, Forum on Information Resource Management, Drake Hotel, Chicago IL. Addressing the theme: "Information Resource Management in the Years of Change," the forum will provide an outlook for the next decade in terms of information systems technology. Contact Ken Burroughs, DBD Systems, 1500 N Beauregard St, Alexandria VA 22311.

June 25-29, Applied Numerical Methods, The University of Michigan, Ann Arbor MI. This course is intended for individuals in industry and branches of the government who wish to acquire a working knowledge of fundamental numerical methods. Emphasis will be placed on computer solutions to practical engineering and scientific problems. Contact Engineering Summer Conferences, 400 Chrysler Ctr, North Campus, The University of Michigan, Ann Arbor MI 48109.

June 27-29, Data Processing Operations Management, Toronto Canada. This seminar will emphasize the management skill and techniques applicable to the data processing operations function. Contact The University of Chicago, Center For Continuing Education, 1307 E 60th St, Chicago IL 60637.

June 27-29, Machine Processing of Remotely Sensed Data, Purdue University, W Lafayette IN. The symposium will focus upon the theory, implementation and novel applications of machine processing of remotely sensed data. Contact Purdue University, Laboratory for Applications of Remote Sensing, 1220 Potter Dr, W Lafayette IN 47906.

July 9-20, Computing Systems Reliability, University of California, Santa Cruz CA. Contact Institute in Computer Science, University of California Extension, Santa Cruz CA 95064.

July 11-13, Microcomputer Applications, Southern Technical Institute, Marietta GA. See August 1-3 for description. Contact Dr Richard L Castellucis, Southern Technical Institute, Electrical Engineering Technology Dept, 534 Clay St, Marietta GA 30060.

July 16-27, Introduction to Digital Electronics and Microcomputer Interfacing, Lexington VA. This hands-on laboratory course is for academic and industrial personnel. There will be approximately 60 hours of laboratory instruction with one microcomputer laboratory station for each two participants. Contact Prof Philip Peters, Dept of Physics, Virginia Military Institute, Lexington VA 24450.

July 19-20, BASIC: A Computer Language For Executives, New York NY. Executive computing will be discussed, including problem solving, planning, forecasting and database systems. Also to be covered are programming fundamentals, the mindless computer, sequence, decision and iteration, computer languages and BASIC. Contact American Management Associations, 135 W 50th St, New York NY 10020.

July 23-27, Finite Element Method In Mechanical Design, The University of Michigan, Ann Arbor MI. This course is intended for engineers working in mechanical design where knowledge of stresses, displacements, or vibratory motion is important. No previous experience with finite elements is assumed. The course will familiarize the attendee with finite element modeling concepts and will review the fundamentals on which the method is based. Contact Engineering Summer Conferences, 400 Chrysler Ctr, North Campus, The University of Michigan, Ann Arbor MI 48109.

August 1-3, Microcomputer Applications, Southern Technical Institute, Marietta GA. The emphasis of this seminar will be on the applications of microcomputers in industry. Software, hardware and interfacing techniques will be discussed. Contact Dr Richard L Castellucis, Southern Technical Institute, Electrical Engineering Technology Dept, 534 Clay St, Marietta GA 30060.

August 6-8, Pattern Recognition and Image Processing, Hyatt Regency O'Hare, Chicago IL. This conference is sponsored by the Machine Intelligence and Pattern Analysis Committee of the IEEE Computer Society. The program will consist of submitted and invited papers and a large trade show of graphics and image processing equipment. Contact PRIP 79, POB 639, Silver Spring MD 20901.

August 8-10, SIGPLAN Symposium on Compiler Construction, Boulder CO. This symposium will consider methods of constructing compilers and experiences with them. The emphasis will be less on theoretical methods and more on techniques applied to real compilers. Contact Prof Leon Osterweil, Dept of Computer Science, University of Colorado, Boulder CO 80309.

August 8-10, First Annual Conference on Research and Development in Personal Computing, Hyatt Regency O'Hare, Chicago IL. This conference is sponsored by the Association for Computing Machinery (ACM) Special Interest Group on Personal Computing (SIGPC). A large trade show of personal computer and graphics equipment is planned to

accompany an assortment of papers, panels, user group meetings, workshops, and person to person poster booths. Contact Bob Gammill, Computer Science Div, Dept of Mathematical Sciences, 300 Minard Hall, North Dakota State University, Fargo ND 58102.

August 13-15, Conference on Simulation, Measurement and Modeling of Computer Systems, Boulder CO. This conference will feature performance prediction techniques employed during the design, procurement and maintenance of computer systems. It will provide a forum for both applied and theoretical work in the disciplines of performance monitoring, modeling, and simulation of computer systems. Contact Gary Nutt, Xerox PARC, 3333 Coyote Hill Rd, Palo Alto CA 94304.

August 13-16, Q-GERT Network Modeling and Analysis, Ramada Inn, Lafayette IN 47905. This course will provide the attendee with the information necessary to model complex systems using Q-GERT. Emphasis will be on the procedures for modeling and analysis. Contact Pritsker and Associates Inc, POB 2413, W Lafayette IN 47906.

August 13-17, High Speed Computation: Vector Processing, The University of Michigan, Ann Arbor MI. In this course, the architectual, software, and algorithmic issues of vector architecture are coordinated through the discussion of concepts in computer architecture, and by detailed study of current vector processors and their use. Contact Engineering Summer Conferences, 400 Chrysler Ctr, North Campus, The University of Michigan, Ann Arbor MI 48109.

August 19-22, International Conference on Computing in the Humanities, Dartmouth College, Hanover NH. This conference is intended to foster computer research and technique in all areas of humanistic study, to promote international cooperation in the development of programs, data banks, and equipment, and to make the results of research available. The program will include a plenary session each evening and shorter sessions during the day. Contact Stephen V F Waite, Kiewit Computation Ctr, Dartmouth College, Hanover NH 03755. ■

In order to gain optimum coverage of your organization's computer conferences, seminars, workshops, courses, etc, notice should reach our office at least three months in advance of the date of the event. Entries should be sent to: Event Queue, BYTE Publications, 70 Main St, Peterborough NH 03458. Each month we publish the current contents of the queue for the month of the cover date and the two following calendar months. Thus a given event may appear as many as three times in this section if it is sent to us far enough in advance.

BYTE's Bits

Call for Papers

The Thirteenth Meeting of the Hawaii International Conference on System Sciences will be held January 3 thru 4, 1980 at the Ilikai Hotel in Honolulu HA. Papers on either theory or practice in the broad areas of computer hardware and software, and advanced computer systems applications may be submitted. Special emphasis will be placed on medical information processing and decision support systems for upper level managers in organizations. The papers will be refereed and printed in the proceedings of the conference. Three copies of the full text of each paper should be sent to Dr Ralph H Sprague Jr, College of Business Administration, University of Hawaii, 2404 Maile Way, Honolulu HA 96822 and must arrive by July 13 1979. Notification of acceptance and those papers to be presented will be mailed by September 15 1979. The conference is sponsored by the Department of Decision Sciences, the Department of Electrical Engineering, and the Department of Information and Computer Sciences at the University of Hawaii in cooperation with the ACM.

Pictures from Space?

People interested in receiving satellite picture images (such as weather maps) would do well to read the 1968 NASA document number NASA SP-5079. The document, entitled *Constructing Inexpensive Automatic Picture-Transmission Ground Stations* by Charles H Vermillion, was possibly the first report aimed at inexpensive receivers. As stated in the report summary:

> This report describes how one can procure or build the antenna, FM receiver, and other components for an Automatic Picture Transmission (APT) ground station. Detailed drawings and parts lists are included. Installation, alignment, and operation of the APT ground station are also described.

When the report was published in 1968, compatible satellites were expected to be operational until 1972 although future extension programs were planned.

The information contained within the report gives enough background to get the serious hobbyist started on a current set-up if updated information can be found. We would be interested in hearing from anyone with current information.

Sixth Colloquium on Microwave Communications Offers Proceedings

The Sixth Colloquium on Microwave Communications was held in Budapest between August 29 and September 1 1978. The organization was undertaken by the Scientific Society for Telecommunication and the Research Institute for Telecommunication of Hungary by sponsorship of the International Union of Radio Sciences and the Hungarian Academy of Sciences. The following principal subjects were on the agenda of the meeting: communication systems — trends and foundations; circuit theory and computer aided design; electromagnetic fields and antennas; microwave circuits and devices; and materials for microwave applications. The proceedings containing all papers have been issued in two volumes of 944 pages in English, and are available at a price of $30 plus postage. Contact OMKDK Technoinform, Budapest, VIII Reviczky u 6 HUNGARY.

Fourth European Conference on Electrotechnics Announcement and Call for Papers

The fourth European Conference on Electrotechnics, EUROCON '80, will be held in Stuttgart Germany on March 24 thru 28 1980. The conference theme is "From Electronics to Microelectronics — Trends and Applications." Key professionals and industry leaders will give a comprehensive overview as well as reports on the latest developments in this area. A commercial and scientific exhibition will illustrate and supplement the technical presentations.

The conference theme will be covered in approximately 160 papers under four main headings: technology of microelectronics; microelectronics in telecommunications and data processing; electronics in electrical power systems and control; and electronics and microelectronics in other fields. Papers are invited to be submitted to: Professor Dr W Kaiser, Chairman Program Committee EUROCON '80, University of Stuttgart, Breitscheidstrasse 2, D — 7000 Stuttgart 1 GERMANY. Abstracts are limited to 500 words and should arrive no later than June 30. The completed text of accepted papers must be received by December 31.

The 3rd World Conference on Medical Informatics Issues a Call for Papers

The 3rd World Conference on Medical Informatics will be held in Tokyo Japan, September 29 thru October 4 1980. Medical informatics is the application of computer technology to all fields of medicine — health care, medical teaching and medical research. The organizers of this conference are seeking papers in clinical care; administrative, educational and public health applications; information technologies and research; and available systems and products. If you are interested in submitting a paper, write to Morris F Collen MD, Chairman of MEDINFO '80 Program Committee, Director, Medical Methods Research, Kaiser-Permanente Medical Care Program, 3700 Broadway, Oakland CA 94611. Final instructions for preparation of papers and special master typing forms will then be sent to you. The deadline for papers is December 10 1979.

Call for Papers: The Eighth World Computer Congress

The Eighth World Computer Congress (IFIP '80), sponsored by the International Federation for Information Processing (IFIP) will be jointly held in Tokyo Japan on October 6 thru 9 1980 and in Melbourne Australia on October 14 thru 17 1980. The Congress will feature presentations on state-of-the-art developments in technology, equipment, and applications prepared by information processing professionals from around the world. In order to identify and schedule these speakers and topics, the Program Committee has recently issued a formal call for papers. Those papers accepted will be delivered in either Tokyo or Melbourne, and in some cases at both locations. Potential authors should contact AFIPS, 210 Summit Av, Montvale NJ 07645 to receive a copy of a brochure which explains all requirements and necessary qualifications.

IEEE Conferences and Meetings

An extensive listing of IEEE Computer Society sponsored conferences and meetings through 1981 is available by writing Harry Hayman, Executive Secretary IEEE, POB 639, Silver Spring MD 20901.

More Cryptographic Notes

Anyone looking for an in-depth article on data encryption should read two papers brought to our attention by William Flynn. One is a reprint of an article by Ehrsam et al which appeared in the *IBM Systems Journal*, Volume 17, Number 2, entitled "A Cryptographic Management Scheme for Implementing the Data Encryption Standard." The cost is 50¢ per reprint and the IBM order number is G321-5066. For $1.75 you can obtain a copy of the issue which is devoted entirely to cryptography. Write to IBM Systems Journal Reprints, Armonk NY 10504.

The other paper, *FIPS Publication 46*, is available from the US Department of Commerce, National Technical Information Service, 5285 Port Royal Rd, Springfield VA 22161, at a cost of $4 for a paper copy and $3 for a microfiche copy.■

Timesharing:

Sheldon Linker
6399 Wilshire Blvd
Suite 1010
Los Angeles CA 90048

Squeezing the Most from your Micro

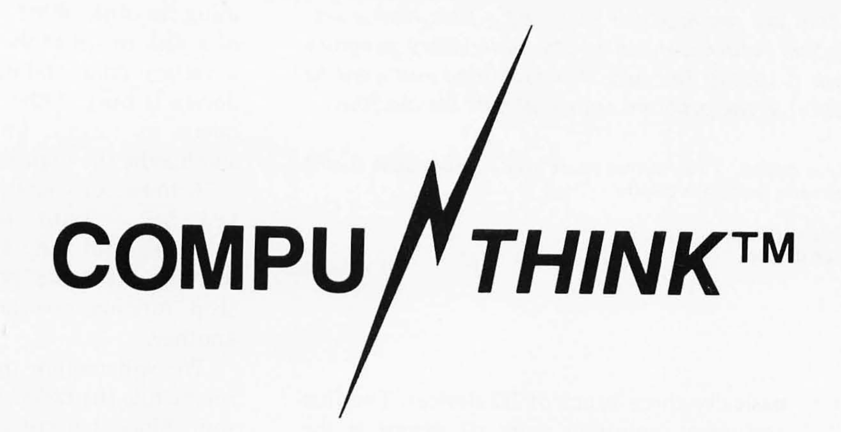
Table 1: An example of a quick relocation scheme designed with a 6800 processor in mind. This set of instructions would be stored along with the program on the auxiliary memory to direct the loader as to how to reinsert the data into main memory each time the program was run. The point of this scheme is to provide a minimal amount of computation when a program is loaded from a library into memory prior to execution. Similar schemes can be chosen for any particular computer's architecture.

Command to Run Time Loader	Explanation
Start absolute loading:	The header code is followed by the absolute start address. In this case, the loader behaves as any other loader. There is no relocation of the data and instructions that follow. Loading starts at the address given.
Start relative loading:	The header code is followed by an address. Loading begins at the first available address, as determined by the operating system. From this point on, a relocation factor will be added to all instructions and data flagged for relocation.
Skip bytes:	This code is followed by a number designating the number of bytes to be skipped. This is useful in defining uninitialized buffers and is more efficient than repeated uses of code to reserve one or two bytes (see below).
Define absolute start address:	The header code is followed by the absolute start address. If the routine is a subroutine, this code would not be used as the module has no start address. When this code is used the program will be started at the specified address once loading is completed.
Define relative start address:	Similar to the preceding code; however, program execution will start in a position relative to the first location.
One byte:	The header code is followed by one byte. This code gets no relocation, because it is either an instruction without an address, or data which is too small to be an address.
Two bytes absolute:	The header code is followed by the two bytes. This code also receives no relocation because it is either an absolute address value, a one byte immediate instruction with its data byte, or it is a relative address instruction which is self-relocating.
Three bytes relative:	The header code is followed by the 3 byte instruction. This code will receive a relocation factor.
Three bytes absolute:	The header code is followed by a 3 byte instruction with an absolute address value which is unchanged in loading.
Two bytes relocatable address values:	The header code is followed by the address data. The address data is always relocatable.
End:	At this point, control returns to the program that called the loader if no starting address was given in the loading module. If the loading module contained a start address that address is called.

Although one normally thinks of timesharing as only working on large computer systems, it is possible to run even on small systems. Many of the newer large scale timesharing systems use virtual memory and swapping, which is not possible or practical on smaller machines. Virtual memory requires mapping hardware (a machine with interruptable instructions, such as an IBM 370). Swapping requires a reasonably fast disk, which will cost at least $2000. What we are left with is an in core system that keeps everything running in real memory at all times.

The first consideration is the assembler and loader. In your current system, a program's location can be assigned only at assembly time. On a timesharing system, the programmer may not know where the program will be located in memory. The reason knowledge of this location is conditional is that a decision point in the design of the system has been reached. If the system is to be nonrelocatable, the programmer may define the location of the program. The problem that arises here is that if, at the time the program is to run, the place in memory that the program was supposed to run in is already occupied, it cannot be loaded. On the other hand, if the system is capable of relocating, the program can be put anywhere in memory. This produces the additional benefit that subroutines do not have to be assembled with the program. To perform this relocation the assembler leaves offset information in the object tape or file which the loader will interpret as it goes. One possible relocation code scheme is shown in table 1. Of course, all sorts of schemes are possible. Note that relocation alone will take some amount of coding and execution time.

The second consideration is the allocation of system resources. In most cases this should concern only IO devices, although there may be some systems with interrupts not associated with IO devices. There are



ATTENTION: COMPUTER DEALERS

Compu/Think invites you to preview our new integrated business micro-computer system at the June NCC in New York. It has to be seen to be believed! System features:

- 108,544 character semi conductor memory
- Resident ROM Operating System
- Resident ROM Microsoft BASIC
- Resident ROM machine language monitor
- Interactive de-bug facility
- High resolution graphics
- Full screen editing including split screen and individual field modes
- Two disk storage versions:
  1) 712,000 character memory
  2) 2,400,000 character memory

all at retail prices of from $4,495.00 to $5,995.00

We will also be showing our software and other peripheral products for the Commodore PET and Radio Shack TRS 80 including dual disk drives.

Location and Dates
Sheraton Center
Booths 130 and 131
New York, New York

Week of June 4th, 1979

Dealer Inquiries Invited
Contact:
Frank R. Price
Director of Marketing
3260 Alpine Road
Menlo Park, CA 94025
(415) 854-2577

"COMPUTER SYSTEMS FOR BUSINESS"

Circle 71 on inquiry card.

Table 2: Minimum routines that are required for handling a timesharing system. The end task routine should return control to the supervisory program with information that the task is totally finished. The last thing you want to do is encounter a halt instruction in the program code and halt the machine.

- Attempt to allocate a particular device. This routine must give a return code stating whether or not the device is already being allocated.
- Free a device.
- Read a character from a particular device.
- Write a character to a particular device.
- Read a particular disk block.
- Write a particular disk block.
- Wait.
- End a task.

basically three types of IO devices. The first and most common type of device is the single owner. This is a device which can only be used by one task at a time. (A task is a program running in the timesharing system.) An example of a device which must be single owner is a cassette recorder. It would just not do to have someone else's data in the middle of your program.

The second type of IO device is the shareable unit. The most common example of this is the floppy disk. For a disk to be correctly shared, the operating system routine which is handling the disk must reposition the heads every time the disk is used. Most systems already use this method, but there are those that have a call to position the head and another set of calls to read, write and verify. Separate calls cannot be used because a second task might reposition the heads before the first task had a chance to read or write.

The third type of IO device is the device that is the system's alone. An example of this is the clock interrupt, a solitary interrupt device. It must be the system's job to keep track of time. It is also the charge of the system to keep track of which devices are owned by which tasks. The system must place all of the task's allocated devices back on the available list if a "cancel the program" function is executed.

When a task wants to perform input or output, it might use a considerable amount of system time monitoring status lines, thereby making timesharing impossible, unless all, or at least some of the devices are interrupt driven. The best way to handle things is to have a routine which will cause a task to wait until an interrupt is received for that task, then let the task handle the interrupt, including polling. So far, the routines required are summarized in table 2. (This is not to say that these are the only routines you will ever need. Table 2 is probably the minimum set of functions you will ever need.)

When handling disk interrupts, it is necessary to keep track of which task, if any, is using the disk. When a task requests the use of a disk or other shared device, it must get a return code stating whether or not the device is busy. Otherwise, the system must queue its request (make the program wait and handle the request whenever it can).

A third consideration is scheduling. Each task has a status: ready to run, running, running with an interrupt pending, or waiting. At some point, the system must stop running one task and begin running another.

We will require the operating system to reschedule the tasks every time a task asks to wait. Since that task cannot proceed, we will perform a task that is not in a wait state. There are three other times when we may optionally reschedule the tasks: every interrupt, every clock interrupt, or every interrupt and system call. These methods are called demand scheduling, event scheduling, time slicing, and quick scheduling, respectively. The fastest method is to wait for WAIT calls. The other three methods are fairer, depending on how you look at things.

The actual method of scheduling leads to another decision point. The scheduler may be foreground-background, round robin, or priority scheduling. Foreground-background is the fastest. In this type of scheduling, the system scans down the list of tasks and runs the first nonwaiting task. When this method is used, the position on the list is the important factor.

Round robin scheduling starts the search for an executable task after the last task running. The search starts at the top of the list when it hits the bottom. This way gives every task its chance to run.

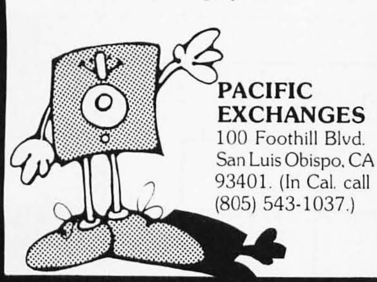
Priority scheduling requires a list of priorities. This scheduler runs the task with the highest priority which is not waiting. This is the fairest method because each task is given exactly what it deserves. When you run off the bottom of the list, using either the foreground-background or priority scheduling method, you have the option of starting over or executing a WAIT instruction. Although it will cost a byte of program memory, it will save considerable time on a 6800 or similar machine, since the interrupt vectoring will be half done by the time you get the interrupt.

The above covers most of what you need, but there are a few more minor considerations:

Creation of tasks: A task has to get into the machine somehow. Two possible methods come to mind. One is the typical timesharing method with each terminal getting its own task. The other is to add a system call which adds a new task.

Memorex Floppy Discs

Lowest prices. **WE WILL NOT BE UNDERSOLD!!** Buy any quantity 1-1000. Visa, Mastercharge accepted. Call free (800)235-4137 for prices and information. All orders sent postage paid.



PACIFIC EXCHANGES
100 Foothill Blvd.
San Luis Obispo, CA
93401. (In Cal. call
(805) 543-1037.)

Circle 295 on inquiry card.

TRS-80 16K MEMORY EXPANSION KIT
INCLUDES 8 TESTED & GUARANTEED MSK 4116 3 16K RAMS. PROGRAMMING PLUGS & EASY-TO-FOLLOW INSTRUCTIONS.
$72.00 PER KIT

6800 64K BYTE RAM SET AND CONTROLLER CHIP SET
MAKE 64K BYTES OF MEMORY FOR YOUR 6800. THE CHIP SETS INCLUDE:
$295.00 PER COMPLETE SET.
32 MSK 4116-3 16K RAMS.
1 MC3480L MEMORY CONTROLLER.
1 MC3242AP MEMORY ADDRESS MULTIPLEXER/COUNTER.
DATA & APPLICATION SHEETS.
PARTS TESTED & GUARANTEED.

16K DYNAMIC RAMS
MSK 4116-3 200NSEC ACCESS TIME/375NSEC CYCLE TIME. TESTED & BURNED-IN.
$8.50 EACH/MIN. QTY. 8

4K STATIC RAMS
EQUIV. TO TMS40L44-30 300NSEC ACCESS TIME/ CYCLE TIME FOR 4MHZ Z-80 OPERATION.
$7.50 EACH/MIN. QTY. 8
TESTED & GUARANTEED

QUANTITY DISCOUNTS AVAILABLE
ALL ORDERS POSTPAID. U.S. FUNDS. CHECK OR MONEY ORDER. VISA, BA, MASTERCHARGE — SEND ACCOUNT NO., EXPIRATION DATE, INTERBANK NO., & SIGNED ORDER.
PHONE ORDERS: 714/633-4460

MEASUREMENT SYSTEMS & CONTROLS, INC.
MEMORY DEVICES DIVISION, DEPT. B3
867 NORTH MAIN ST., ORANGE, CA 92668

Circle 216 on inquiry card.

DATA PROCESSING SUPPLIES

Std. or Mini Floppy . . 2.98 ea.
TRS-80 Cassette . . . . 1.49 ea.
ANSI (notch) Cassette . . . . 4.90 ea.
Qume or IBM 6240 Ribbon . . . . 29.95 dz.

OTHER SUPPLIES AT SIMILAR BARGAINS

Order Now from
John Richards
KEY SUPPLY CO.
2101 S. IH 35 Suite 300
Austin, Texas 78741
(512) 443-1347

All products from major manufacturers, fully guaranteed. No limit while quantities last!

Circle 183 on inquiry card.

Don't Forget!

Our New 4K Byte Non-Volatile Memory Boards Won't Let You!

- 30 days minimum guaranteed data retention
- Ultra low power 450 NSEC static CMOS RAM IC's
- On-board regulator, power monitor and battery
- S-100 bus compatible

Assembled and Tested
$395.00

Remember . . .
to send for details!

CYGOL
1395 Golf Street
Dayton, Ohio 45432

Circle 88 on inquiry card.

SURPLUS ELECTRONICS

ASCII ASCII

IBM SELECTRIC® BASED I/O TERMINAL WITH ASCII CONVERSION INSTALLED $645.00

- Tape Drives • Cable
- Cassette Drives • Wire
- Power Supplies 12V15A, 12V25A, 5V35A Others, • Displays
- Cabinets • XFMRS • Heat Sinks • Printers • Components

Many other items
Write for free catalog
WORLDWIDE ELECT. INC.
130 NORTHEASTERN BLVD.
NASHUA, N.H. 03060
Phone orders accepted using VISA or MC. Toll Free 1-800-258-1036
In N.H. 603-889-7661

Circle 391 on inquiry card.

HAZELTINE 1400

only
$649.95!



- **Verbatim Mini Diskettes . . . . . . $3.70 each (boxes of 10)**
- **Intertube . . $784.00**
- **TRS-80 16K Level II Expansion Kit $89.95**
- **Centronics 779 tractor $1050.00**
- **Horizon II ass. . . . $1999.00**



Mail Order Only.

TORA SYSTEM INC.
29-02 23rd Avenue
Astoria NY 11105
(212) 932-3533

Circle 371 on inquiry card.

6800 DEVELOPMENT PAC II

- 6800 MICRO COMPUTER
- 48K RAM
- EROM PROGRAMMER
- RS-232 INTERFACE
- CASSETTE INTERFACE
- MONITOR DEBUG ROM
- EDITOR/ASSEMBLER
- 4K INDUSTRIAL BASIC
- RACK, BACK PLANE, SUPPLY
- $1895

WINTEK Corp.
317-742-6802
902 N. 9th St., Lafayette, IN 47904

Circle 389 on inquiry card.

CASSETTE AND FLOPPY DISC LABELS.





Avery offers a complete line of labels for cassettes and floppy discs—all with removable adhesive. Including these standard sizes:
Cassette Label (1⅝" x 3½") #5250
Floppy Disc Label (1¾" x 5½") #5252
Write for more information and free samples to:
Avery Label
777 East Foothill Blvd.
Azusa, CA 91702



An Avery International Company

Circle 20 on inquiry card.

RT/68 - M6800 Real Time Operating System

— Executive controlled, event driven
— up to 128 fixed priority tasks run concurrently
— intertask communication through messages
— Executive requires only lk bytes
— NCR task: monitor Console Functions
— CLOK task: task timer, time-of-day, calendar
— KBD task: input from ASCII keyboard
— CRT task: output to ASCII display
— TTY task: input/output from ASCII terminal
— Proven in Industrial Supervisory/ Control applications

Sources on diskette $350.00

Motorola MDOS Format at Sierra Digital X6800 format

Broadcast Computer Systems
323 Penworth Way S.E.
Calgary, Alberta Canada T2A 4G1

Circle 34 on inquiry card.

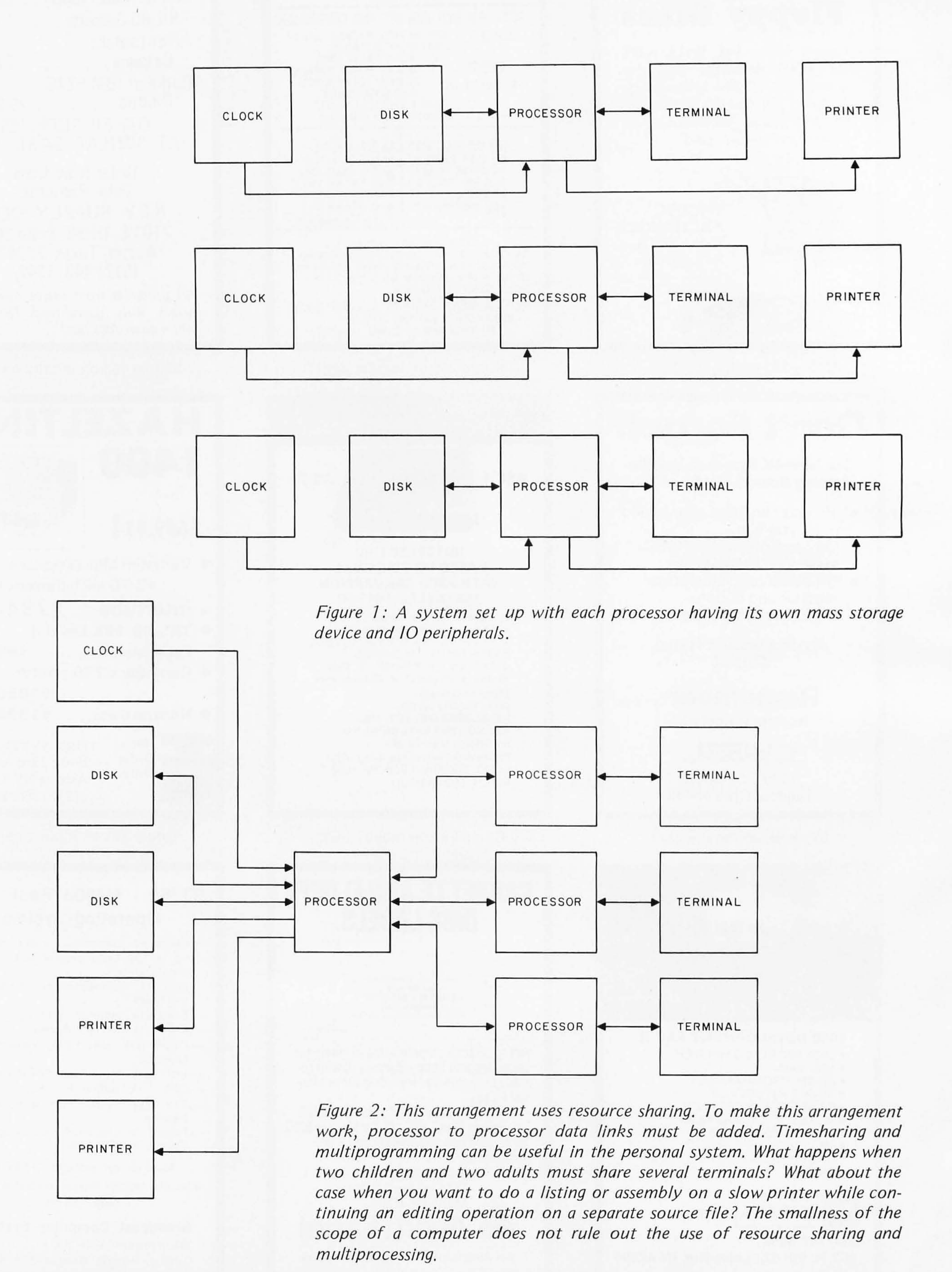


Figure 1: A system set up with each processor having its own mass storage device and IO peripherals.

Figure 2: This arrangement uses resource sharing. To make this arrangement work, processor to processor data links must be added. Timesharing and multiprogramming can be useful in the personal system. What happens when two children and two adults must share several terminals? What about the case when you want to do a listing or assembly on a slow printer while continuing an editing operation on a separate source file? The smallness of the scope of a computer does not rule out the use of resource sharing and multiprocessing.

Memory: You can set things up so that each task has a fixed amount of memory (which may or may not be reset between tasks) or use some sort of a system where the tasks can acquire and free memory dynamically.

Niceness: Programs must be nice to one another, as very few of the machines around have any sort of memory protection or privileged instructions.

Machine considerations: When an interrupt occurs, or a task is otherwise stopped, the registers, including the program status word (PSW), and stack pointer must be saved and later restored. Depending on the type of programs you run and your type of machine you may have to save and restore all or part of page zero. If you have a 6502, you will also have to deal with the stack's page.

Reentrancy: Programs which can be run concurrently by more than one task are reentrant. You may wish to set up some way of effectively using reentrant programs, such as having a null task, into which may be put reentrant subroutines; or by having various small reentrant routines always in the same place in memory, such as multiply and divide.

There are other methods of going about this completely, which I mention only in passing. Many BASIC systems will have one BASIC interpreter in memory along with multiple programs, and will execute one line of BASIC code and then go on to the next pseudotask. This will also work for APL, although long matrix operations will tend to extend the intervals between transitions from one process to another. (Of course, it is a debatable point whether or not a timesharing APL and two workspaces will ever fit into the same memory at one time.)

Multiple processor timesharing systems are also possible. Assuming that you have a central processor with disks and printers, there is a method that can save a lot of money. This method is resource sharing. Figure 1 shows a typical group of three computers each working independently. Each processor handles everything with inefficient use of the printers and disks. Figure 2 depicts a resource sharing setup. This requires the addition of processor to processor data links. In this setup, each peripheral processor does the computing while the central processor handles queued IO and interrupts much like the simple timesharing systems above.■

About the Author

C Brian Honess is currently assistant professor of management science in the College of Business Administration at the University of South Carolina. He reports that he has been an active "building" radio amateur (ham) for 20 years; his interest in computers goes back to programming scientific business applications on an IBM 1620. He learned about what was inside computers by buying a surplus IBM 704 from the government, and slowly taking it apart (donating, selling and scrapping the parts as he went). Another 704 was eventually purchased, and this time it was built back up, from the inside out. This is not exactly a typical personal computer.

Three Types of Pseudorandom Sequences

C Brian Honess
Asst Prof of Management Science
College of Business Adm
University of South Carolina
Columbia SC 29208

Random numbers are extensively used in virtually all areas of data processing, from the simplest games for a hobby microprocessor, up to the most complex business and scientific applications. Deterministic games programmed without the benefit of some random parameter soon become boring and easy to "beat," so it would seem that random number generation and testing should be of interest to even the neophyte programmer or computer hacker when trying to get a simple game up and running. Random numbers are used extensively in various business applications. For example, random numbers would be used by an auditor faced with a large number of transactions to audit, and using a sampling technique to only look at a certain percentage of representative transactions. The number of checkout stations at your local discount department store may have been determined by using a mathematical model of the store, wherein the arrival and departure of "customers" was simulated using random numbers. Market research makes extensive use of random numbers, in selecting the people, streets, blocks, households, etc, to interview or to mail questionnaires. A mathematical model can also be "built" of an element, molecule or compound, and a particle introduced at random and collisions counted. Suppose further that you had a photograph of some obscure planet, covered in an extremely irregular way with areas you assumed to be water. You could divide the photograph into small squares, or maybe overlay with a piece of graph paper, and then "take shots" at the grid with a random number generator, wherein the random number would determine the coordinates of the "shot" and you could then tally the number of "hits" and "misses" and thereby determine the number of hits out of the total number of shots, and get an approximation of the percentage of the surface covered by water.

Before reading on, let me suggest that you try a short experiment. Consider the set of integers from 0 to 99, and quickly write down a list of random 2 digit numbers. Use whatever your current idea of random is, and make a list of 100 numbers. Later we'll see several methods for determining how random your numbers are, but I'll hasten to guess that they won't be very random. Psychologists repeatedly show that the average human just cannot think up random numbers. Upon inspection, there might be too many 4s compared with 6s, or maybe very few 0s and an abundance of 5s.

While it's true that a machine can produce a much better selection of truly random numbers than a human, the problem is that the numbers produced by the machine aren't really random either. If you could build a perfect roulette wheel, you'd get truly random numbers, but the mechanical considerations of such a device are, of

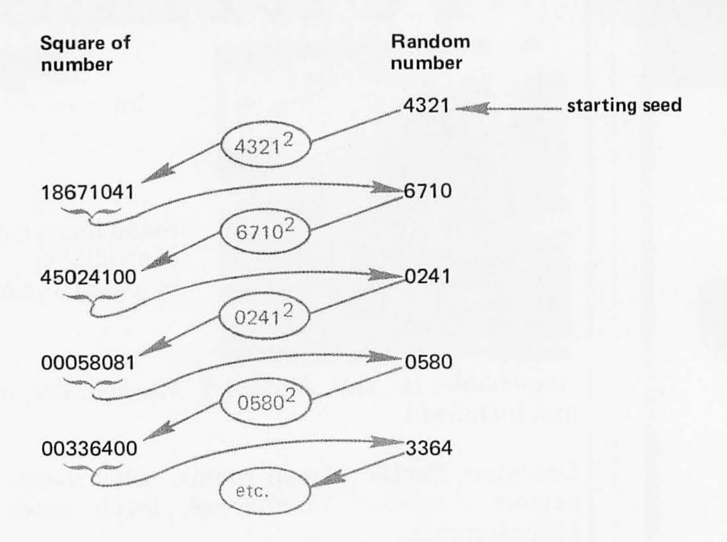


Figure 1: Operation of the center squared method of random number generation can be illustrated by this example. Here we use 8 digit decimal numbers, and assume the ability to extract the center four digits as the 4 digit random number for each cycle. [In principle this algorithm could be done on a binary basis by picking the middle 16 bits of a 32 bit product, or the middle 32 bits of a 64 bit product . . . CH] *All versions of this algorithm are subject to the problem of degeneration, since if the middle digits happen to become 0, the square will continue to be 0 through successive generations of the algorithm.*

course, impossible, to say nothing of the costs, speed, maintenance, testing, and so forth. There is really not much need to strive for such a device for the usual application of random numbers, because there are some mathematical methods which produce what are called pseudorandom numbers. Implemented on a computer, they are quite fast, easy to implement, and just as much fun to play with as a roulette wheel!

Before looking at some of these methods, I might mention that there are a couple of other ways to get random numbers for your games, experiments, or business applications. You could always punch or key into your system as many numbers as you want from the Rand table. This is a formal table used by statisticians entitled *A Million Random Digits* and published by Rand Corporation. There are a million of them, so this could take quite a while. Of course, you're assured that these numbers are thoroughly tested and as unbiased as possible, but assuming you have the time and perseverance to do the job, unless you work out some scheme for using different parts of the table or different orderings, you'll always get the same string of digits. You might find an abbreviated table in the back of some statistics book and use the numbers therein, but the problem here is that you'll probably need more numbers than appear in the table. Of course you could always go through the table more than once, but this doesn't multiply the size of the table. In some applications it might be desirable to be able to use the same random numbers, in the same order, more than once. For example, you may want to duplicate the results of an experiment, an audit, a market research test, or a game. But usually, you'll want a new string of numbers, and this can be secured by selecting starting values or other parameters in the mathematical algorithms that follow.

BASIC Program	Sample Results
100 PRINT "INPUT A 4-DIGIT SEED";	
110 INPUT N	4321
120 LET N = N * N	18671041
130 LET N = N / 100	186710.41
140 LET N = INT(N)	186710
150 LET A = A / 10000	18.6710
160 LET A = INT(A)	18
170 LET A = A * 10000	180000
180 LET N = N – A	6710
190 PRINT N	6710
200 END	

Listing 1: A BASIC program which accomplishes one generation of the center squared method of calculation. Note that this program assumes an interpreter with at least eight decimal digits of accuracy.

The Center Squared Method

The earliest computer oriented method for producing pseudorandom numbers was probably the center squared method. In this method we begin with a *2n* digit number, square it, and then extract the center *2n* digits from the *4n* digit result, and this becomes the next random number, and also becomes the number which is squared in the next iteration. For example, suppose you want some 4 digit random numbers. In this case, of course, $n = 2$, and let's assume we start off with the number 4321 as our "seed" value. Figure 1 shows the process through three iterations.

This method makes a good little program to assign to a beginning programming class because it is easy to explain, easy to determine what the answers "should be," but it has several problems which arise as you get deeper into the problem. In FORTRAN or BASIC, lacking any specific digit manipulation instructions, the hard part comes when you try to strip off the digits either side of the center. Listing 1 shows a simple BASIC program which will generate one random number. *[This program assumes an interpreter with greater than eight digits of arithmetic precision]*. Here, we see that we desire four digits, and enter the seed 4321. Squared,

NEED EIA CABLE?
Data Set has it!

They don't call us "The Company with a Lot of Connections" for nothing! Whatever your interface cable needs, call on Data Set. We have all types of cable and connectors, pinned to your specifications – ribbons, coax, twin-ax, Telco, junction boxes . . . to name just a few.

THESE ITEMS IN STOCK ▶ EIA RS 232-C Cables – All 25 Pins Wired MALE-to-MALE or MALE-to-FEMALE

Specify when ordering:

3-foot length **$17.50 ea.** ● 5-foot length **$18.50 ea.** ● 10-foot length **$21.00 ea.**

SPECIAL TO JULY 15, 1979 ONLY!
10% Discount on any of the above items.
PLEASE MAKE PAYMENT WITH ORDER.

New catalog available



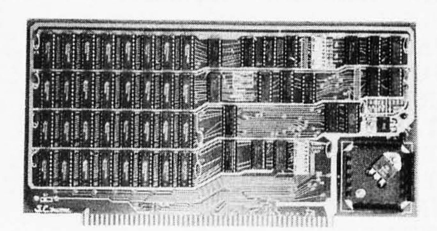
The Company with a lot of Connections
Data Set Cable Company, Inc.
We serve you from 2 locations:

722 Danbury Road — P.O. Box 622
Ridgefield, Connecticut 06877
(203) 438-9684

3660 Polaris Avenue, #21
Las Vegas, Nevada 89103
(702) 873-2133

Circle 331 on inquiry card.

16K STATIC RAM



Kit price
$285 450 nsec
$320 250 nsec
Memory Chips

Add $25 for assembled, tested, Guaranteed.

Static TMS 4044- Fully Static 4Kx1 Memory chips for better data integrity and DMA compatibility.
Fully S-100 Bus Compatible-All lines fully buffered, Dip Switch Addressable in two 8K blocks, 4K increments. Write Protectable in 2 blocks, Memory Disable using Phantom (pin 67) or strappable to any other pin.
Bank Select-Using Output port 40H (Cromemco software compatible)-addressable to 512KB of RAM or for time share. Also has alternate port 80H-making over 1 million byte of RAM available.
Quality Components-First quality parts, fully socketed. Glass epoxy board with silk screened legends, solder masks, Gold Contacts.
Guaranteed-parts and labor for 1 year. You may return the undamaged board within 10 days for a full refund. Foreign and kit purchasers-parts only guaranteed; no return privilege.
Orders- You may phone for Visa, MC, COD orders. ($4 handling charge for COD orders only) Personal checks must clear prior to shipping. Shipping-Stock to 72 hours normally. Will notify expected shipping date for delays beyond this. Illinois residents add 5% tax. Please include phone number with order.

S.C. Digital P.O. Box 906, Aurora, IL 60507 Phone: 312-897-7749

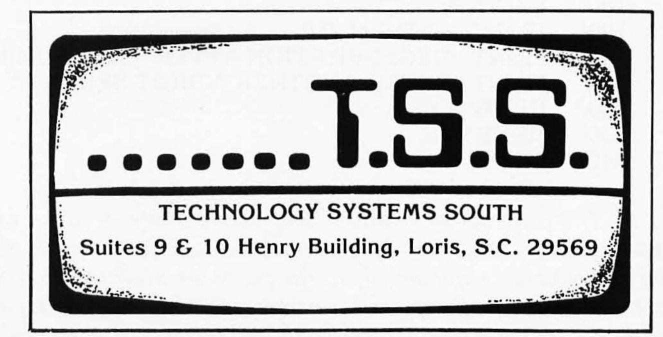
WE HAVE IT

What is IT? The New Pascal Microengine.

IT features the new Western Digital 16 Bit Pascal Microengine CPU, 64K RAM, (2) RS232 ports, (2) parallel ports & (2) Shugart 8014 dual density drives. IT will directly execute Pascal generated P-Code (the only CPU on the market designed to directly execute a high level language), & IT will run compiled BASIC. Wait, that's not all — IT normally sells for **$4,495**, but you can have IT for **$4,195**. Dealer prices are much lower. Plus we have software galore.

Call us about IT at (803) 756-6000.

THE PASCAL MICROENGINE
FROM THE COMPANY THAT GIVES YOU TOMORROW'S INVENTIONS TODAY



Circle 364 on inquiry card.

```
100   PRINT "HOW MANY NUMBERS DO YOU WANT?";
110   INPUT J
120   PRINT "INPUT A 4-DIGIT SEED";
130   INPUT N
140   FOR I = 1 TO J
150   LET N = INT(N*N/100)
160   LET A = INT(N/10000)*10000
170   LET N = N - A
180   PRINT N
190   NEXT I
200   END
```

Listing 2: A BASIC program which will generate a list of random numbers using the center squared method. The program embeds a revised form of the calculation of listing 1 within a FOR-NEXT loop.

this becomes an 8 digit number in line 120. Line 130 divides by 100 in the first step of several necessary to get out the four center digits. Of course if you wanted 2 digit random numbers your seed would have been a 2 digit number; you'd have a 4 digit square, and you would change line 130 to divide by 10. For 6 digit numbers, you'd change it to divide by 1000, etc. Line 140 completes the removal of the right-hand two digits by integerizing the number. This resulting 6 digit number is then set aside, and you start removing the front two digits. This is done by dividing by 10000, integerizing the result and then multiplying by 10000. In line 180 you subtract this number from the one previously set aside and out come the center four digits, the new random number. If you are working with 2 digit random numbers, the divisor in line 150 would be changed to 100 and you'd then multiply by 100 in line 170.

One number isn't going to be enough for most applications, so let's put in a loop and get "n" numbers. Listing 2 shows the modifications necessary. Also, we'll combine lines 120 to 140, and lines 160 to 170 in listing 1.

I remember when I first coded this method in a beginning FORTRAN class. I've forgotten what 4 digit seed the instructor used at the time, but it was a revelation when I found out about something he called "degeneration." I'm sure a simple program can be written to discover any and all of the 4 digit seeds which will cause this program to degenerate to zero, but let's assume that there is at least one, and that Murphy's Law will guarantee that this particular one is the seed you choose for your first run. It is not difficult to imagine that there is a 4 digit number, which, when squared, will have four zeros in the middle. Maybe your number squared will be 12000034, or 65000025, etc. This being the case, you'll get 0000 as your next random number until you discover what is going on and get out of the loop. Listing 3 shows how we'll test for that problem and perform an ad hoc fix; we'll just call for another seed when a random number of zero is obtained.

Our final try at the center square program still doesn't solve one of the worst problems with this method. The method doesn't give very long periods for many seeds, and you really can't predict what the results will be until you try it. The method starts repeating numbers, and even the place where it starts repeating can't be determined without trying it. For example, you may print out 722 different numbers, and then it will start repeating the last 34 of them. The method is easy, and it is fun, and it may just produce all the pseudorandom numbers you need for your application.

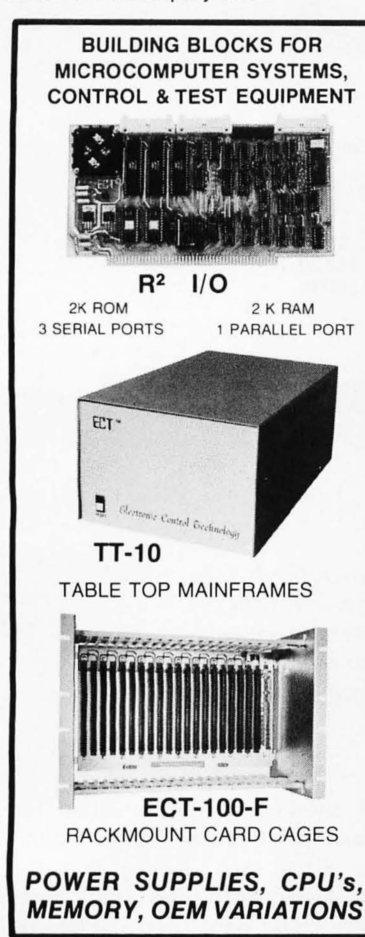
Fibonacci Series Technique

A second method for generating random numbers makes use of the Fibonacci series, so named for its discoverer Leonardo of Pisa, known as Fibonacci (meaning son of Bonaccio). Leonardo was perhaps the greatest European mathematician of the Middle Ages, and if not for him you might be programming your machine using Roman numerals, because it was Leonardo who recognized the enormous superiority of the Hindu-Arabic decimal system with its positional notation and the zero symbol, over the much clumsier Roman system. Table 1 shows several numbers in the Fibonacci series, and you'll notice that each number is simply the sum of the previous two numbers. Actually, we could make up any number of series by starting with any

```
100   PRINT "HOW MANY NUMBERS DO YOU WANT?";
110   INPUT J
120   PRINT "INPUT A 4-DIGIT SEED";
130   INPUT N
140   FOR I = 1 TO J
150   LET N = INT(N*N/100)
160   LET A = INT(N/10000)*10000
170   LET N = N - A
180   PRINT N
190   IF N <> 0 THEN 230
200   PRINT "DEGENERATION AFTER"; N ; "NUMBERS"
210   PRINT "ENTER ANOTHER 4-DIGIT SEED";
220   INPUT N
230   NEXT I
240   END
```

Listing 3: The program of listing 2 will occasionally produce examples of degenerate cases. The center squared method is prone to such degeneration with an unpredictable frequency, so for purposes of illustration this version incorporates an ad hoc fix to ask for a new seed when degeneracy is detected, and report on how many cycles were required to reach degeneracy.

Circle 120 on inquiry card.



POWER SUPPLIES, CPU's, MEMORY, OEM VARIATIONS

SPECIALIZING IN QUALITY MICRO-COMPUTER HARDWARE

INDUSTRIAL
EDUCATIONAL
SMALL BUSINESS
PERSONAL

ELECTRONIC CONTROL TECHNOLOGY

763 RAMSEY AVE.
HILLSIDE, N.J. 07205

(201) 686-8080

Computer Lab of New Jersey

Computer Lab sells the best S-100 Bus products at the best possible prices. Not only are our prices great, so is our delivery. We offer a 10% discount on most major lines, plus a 5% additional discount for a cash purchase.

	LIST PRICE	OUR CASH PRICE
Godbout 8K Unkit	$139.00	$118.00
Godbout 16K Unkit	$295.00	$252.00
Thinker Toys Super RAM 16K Kit	$299.00	$255.00
Vector Graphic 8K Assembled	$245.00	$209.00
Problem Solver 16K Assembled	$429.00	$366.00

Subject to available quantities. Prices quoted include cash discount.
Shipping and Insurance Extra
Cut-off Date July 15

Call for our prices on:
Cromemco, Dynabyte, Godbout, Integral Data Systems, IMC, Ithaca Audio, Meca, Micropolis, Mountain Hardware, Mullen, Oliver, SSM, Soroc, Tarbell, TEI, Teletek, and many other fine products.

Computer Lab of New Jersey
141 Route 46 • Budd Lake, N.J. 07828
Phone: (201) 691-1984

HOURS: Monday & Friday: 12 to 6, Tuesday-Thursday: 12 to 9
Saturday: 10 to 5

Call or write for our free catalog & price list

MONITOR $139, 12" B&W



COMPLETE KEYBOARD TERMINAL $350
Full Kit $295 (includes SCT-100)

Stand Alone Video Terminal

!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~

SCT-100 VIDEO BOARD FEATURES
- 64x16 line format with 128 displayable characters
- Serial ASCII or BAUDOT with multiple Baud rates
- **$187** Assembled or **$157** Kit (Partial Kit **$95**)
- Full cursor control with scrolling and paging
- On board power supply
- Serial interface RS232 or current loop
- Purchase SCT-100 alone or complete terminal

XITEX CORP. 13628 Neutron P.O. Box 402110
Dallas, Texas 75240 (214) 386-3859

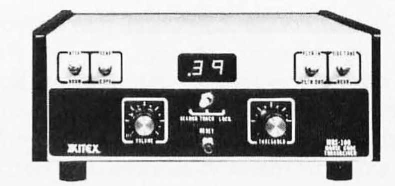
ORDER BY PHONE/Overseas orders & dealers welcome

masterchange **XITEX MORSE TRANSCEIVER** VISA

• $95 Partial Kit • $225 Complete Kit • $295 Assembled

SEND
- 1 to 150 WPM (set from terminal)
- 32 character FIFO buffer with editing
- Auto Space on word boundries
- Grid/Cathode key output
- LED Readout for WPM Buffer space remaining

COPY
- 1 to 150 WPM with Auto-Sync.
- Continuously computes and displays Copy WPM
- 80 HZ Bandpass filter.
- Re-keyed Sidetone Osc. with on-board speaker.
- Fully compensating to copy any 'fist style'



SERIAL INTERFACE
- ASCII (110, 300, 600, 1200) or Baudot (45, 50, 57, 74) compatible
- Simplex Hi V Loop or T2L electrical interface
- Interfaces directly with the XITEX® SCT-100 Video Terminal Board; Teletypes® Models 15, 28, 33, etc.; or the equivalent

Circle 400 on inquiry card.

(a) **The Fibonacci Series**

n	F_n
1	1
2	1
3	2
4	3
5	5
6	8
7	13
.	.
.	.
.	.
19	4181
20	6765
21	10946
22	17711
.	.
.	.
.	.

(b) **Some Characteristics of the Fibonacci Series**

Final digit (LSD)	repeats	in	cycle	of	60
Last 2 digits	repeat	in	cycle	of	300
Last 3 digits	"	"	"	"	1500
Last 4 digits	"	"	"	"	15000
Last 5 digits	"	"	"	"	150000
etc.					

Every 3rd F_n is	divisible by	2		
" 4th "	"	" 3	}	Note: this is also the Fibonacci series.
" 5th "	"	" 5		
" 6th "	"	" 8		
etc.				

Table 1: The Fibonacci series is a numerological phenomenon which is generated by the following definition: the next term in the series is the sum of the previous two terms, with the first two terms defined to be a value of 1 as a starting point. At (a) are listed several representative sections of the Fibonacci series, and at (b) are shown several miscellaneous characteristics of the Fibonacci series abstracted from the mathematical literature. This series can be used as a basis for a random number generator, as described in the text.

two numbers and letting their sum be a new random number, and repeating this for our desired number of iterations. The reason that we'll use the Fibonacci series specifically, and not any other series, is that the characteristics of the Fibonacci series have been studied, and we know several facts about it that will be of interest. Table 1 for example shows that if we want more than 60 single digit random numbers, the Fibonacci series isn't going to work. Of course we could start extracting 2 digit numbers from different parts of the numbers produced, but here we're on our own as far as statistical characteristics are concerned.

```
100   PRINT "HOW MANY NUMBERS DO YOU WANT?";
110   INPUT J
120   LET A = 10946
130   LET B = 17711
140   FOR I = 1 TO J
150   LET N = A + B
160   IF N < 100000 THEN 180
170   LET N = N - 100000
180   PRINT N
190   LET A = B
200   LET B = N
210   NEXT I
220   END
```

Listing 4: A BASIC program which implements a Fibonacci series random number generation technique. The program works machines of finite precision (even though the Fibonacci numbers eventually get infinitely large) because only the low order digits are kept as part of the pseudorandom number. Since the high order portion of a Fibonacci number has no effect on the low order portion during calculation of the next number, it is possible to completely ignore the high order part.

Listing 4 shows a BASIC program for calculating and printing "n" random numbers of five digits each. Notice that the generator is seeded with two seeds from table 1. These could have been INPUT, of course, and in that way a different series of random numbers could be produced. I've chosen the first two 5 digit numbers in the sequence, but there is nothing special about them. Also, you might consider having the generator run through the loop a number of times before it starts printing the output. This could be easily implemented with another INPUT statement and another FOR . . . NEXT loop, or maybe by just adding the number of unwanted numbers to J, and then putting in an IF to suppress printing of the first J-N numbers. Listing 4 is straightforward: after determining how many numbers you want, it takes the two seeds and calculates the first number. It is possible that the result will be over five digits when the two previous numbers are added, but it can never be greater than 199998 (99999 + 99999), so we check for this condition in line 160 and simply subtract 100000 if the number is larger than 99999. Lines 190 and 200 serve to shift the second number into the location previously holding the first number, and the new random number into the location previously holding the second number, and we're ready for a new iteration.

In order to find the n^{th} Fibonacci number, you needn't go up to "n" one at a time. There is an easier method, although you might not think so when you see the formulas in figure 2. You might try to find the 20^{th} Fibonacci number with your

(a) $$F_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$$

(b) $$F_n = \frac{\Phi^n}{\sqrt{5}} = \frac{((1+\sqrt{5})/2)^n}{\sqrt{5}}$$ rounded to nearest integer.

(c) $$(\text{Golden ratio})^k = \left(\frac{1+\sqrt{5}}{2}\right)^k = \Phi^k = 1.61803^k \ldots$$

Figure 2: The program of listing 4 was an iterative calculation. It turns out that there are several ways to calculate Fibonacci numbers directly which do not involve iteration. Formula (a) is an exact calculation of the n^{th} Fibonacci number. Formula (b) is also an exact calculation if the result is rounded to the nearest integer. Formula (c) defines a criterion for making the Fibonacci sequence pass various statistical tests which would otherwise fail: pick every k^{th} number where k is chosen so that the "golden ratio" to the k^{th} power is relatively large compared to the low order portion of the Fibonacci numbers which is used as a random number output.

pocket calculator (or maybe your computer?) using formula a of figure 2. We know the answer is 6765, from table 1. This formula produces the exact answer, but we can get it with a little less calculation by using formula b in figure 2. The term $(1 + \sqrt{5}) / 2$ is known in the mathematical literature as the "Golden Ratio" and is often symbolized by the Greek letter Φ (phi). If formula b is evaluated and then rounded to the nearest integer, it will produce F_n. You might try this, again, with n = 20.

The Golden Ratio assumes importance when using the Fibonacci series random number generator, because it is used as a "correction factor." The results of the generator, aside from being somewhat predictable as shown in table 1, fail many of the statistical tests usually applied to random number generators. A big improvement can be made in the results if we use only every k^{th} number, where k is almost any number which will make the Golden Ratio to the k^{th} power relatively large. Figure 2c is the required formula. If this modification is implemented, and k is large, your calculating time for each random number that is to be used will greatly increase, but you'll have numbers that are about as good statistically as any other method.

Power Residue Calculations

A third general class of pseudorandom number generators is called the Power Residue Method. It is this method that is usually favored by hardware manufacturers, software writers and mathematicians, because long periods prior to repetition can be assured, and the numbers generated hold up well to statistical tests for randomness. The method is, however, machine dependent since it relies on the word size of the machine. The Power Residue Method is the method employed in RANDU, an extremely popular random number generator appearing in the "Scientific Subroutine Package" (IBM publication number H20-0205) for the IBM System 360 and 370 computers. The publication gives a FORTRAN listing of this subroutine and documentation on how to use it, and also delineates a FORTRAN listing and instructions for use of GAUSS, which is a program for producing a normal distribution of random numbers. The methods can easily be extended to distributions other than the normal. Background on the number theory aspects of the Power Residue Method can be obtained in another IBM booklet, "Random Number Generation and Testing" (IBM publication number (C20-8011).

```
100  PRINT "INPUT ANY ODD INTEGER";
110  INPUT N
120  LET X = N * 65539
130  LET Y = X * 0.4656612873077392578125E-09
140  PRINT X ; Y
150  END
```

Listing 5: A BASIC program to calculate one cycle of a pseudorandom sequence using the power residue method. This particular program is the algorithm used for a 32 bit machine as found in the IBM System 360 and 370 "Scientific Subroutine Package," IBM Publication Number H20-0205. In the source document cited, this algorithm is given as a FORTRAN subprogram named RANDU.

Listing 5 shows a BASIC version of a program to produce one random number on a 32 bit machine. The program can be easily modified, of course, along the lines we followed for the center squared and Fibonacci methods covered earlier. The multiplier in line 130 is 2^{-31} and of course you'll be rounding it to fit your particular BASIC compiler. Line 130 simply transforms our new random number X into a floating-point version between 0 and 1, which is a more usual way of delineating random numbers. With a 32 bit machine (1 sign bit) we use 2^{-31}, and this would be changed to correspond to the particular machine upon which the method is implemented. The multiplier in line 120 is also machine dependent. It has the form: $8i \pm 3$, where i is any integer. The trick here is to choose i, such that the resultant multiplier is close to $2^{b/2}$. Since $b = 32$ for this example (b is the number of bits), then we

want the multiplier to be close to 2^{16} = 65536. If i = 8192, then the multiplier will be 65536+3=65539. The seed multiplied by the multiplier produces a product which is $2b$ bits long and we discard the b high order bits, and the remaining b low order bits become the random number and the input seed for the routine for the next number. Using this method we will get 2^{b-2} terms before repeating. Actually, the sign bit doesn't count, so we'll have $2^{31} - 2$ or 2^{29}, or over half a billion numbers before repeating. I'm not about to try and prove this, but I will give it a "go" with a smaller machine assumed.

Let's assume a 6 bit machine. This should produce 2^{b-2}, or $2^4 = 16$ numbers before repeating, and that shouldn't be too difficult to inspect manually. We want a multiplier of the form $8i \pm 3$ which is close to $2^{b/2} = 2^{6/2} = 2^3 = 8$. If $i = 1$, we'd have $8 X 1 + 3 = 11$ and $8 X 1 - 3 = 5$. Both of these possibilities are equally 3 away from our desired value of 8, so let's try both. Table 2a shows how we get started using 5 as the multiplier, and table 2b shows the whole cycle of all 16 numbers produced. Table 2 also shows that if we had chosen a multiplier of 11 the procedure would also have produced 16 numbers before repeating.

You've probably noticed that the two columns of numbers in table 2 just don't look too random. Both columns have numbers that always end in 1. For the 11 multiplier case, the 4th digit is always 0, and the 5th digit alternates between 0 and 1. For the 5 multiplier case, the 5th digit is always 0 and the 4th digit alternates between 0 and 1. Obviously the low order bits are far from random. If you wanted random digits, and not random numbers, it would obviously be to your advantage to choose high order bits, or possibly the bits you discarded when you cut the product from 12 to six bits. The usual scheme, after developing the numbers in table 2, would be to place the binary point at the beginning of the 6 bit numbers, and thereby transform the whole list to a distribution between 0 and 1.

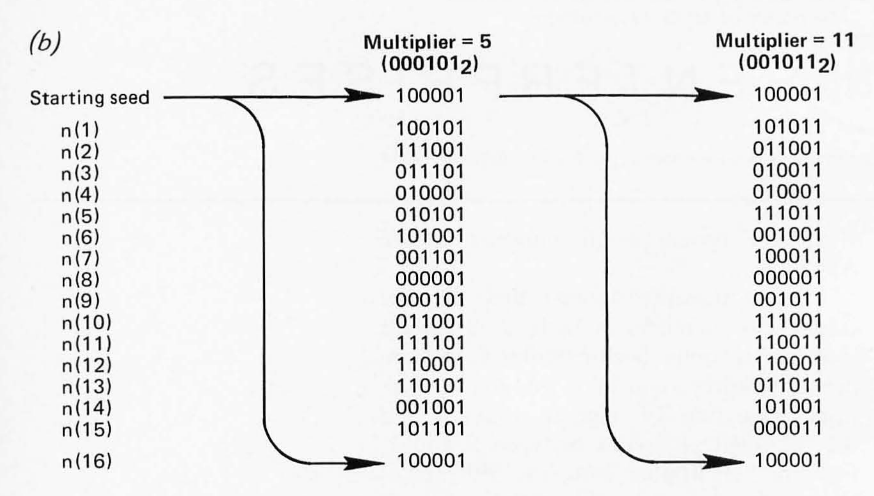
Testing Randomness

Tests of the randomness of a series of numbers usually fall into one of two major categories, those that examine the digits appearing in the numbers and those that

(a)

1. Choose an odd integer starting value. We'll choose the 6 bit number 100001 (simply because it'll be easy to multiply).
2. Choose the multiplier. (We've already decided on 5, binary 101.)
3. Compute the product. (100001) * (000101) = 000010100101
4. Cross out the first six bits, and you have the new number.
   ~~000010~~100101 = 100101
5. (100101) * (000101) = ~~000010~~111001 = 111001
6. (111001) * (000101) = etc.

(b)



	Multiplier = 5 (0001012_2)	Multiplier = 11 (001011_2)
Starting seed	100001	100001
n(1)	100101	101011
n(2)	111001	011001
n(3)	011101	010011
n(4)	010001	010001
n(5)	010101	111011
n(6)	101001	001001
n(7)	001101	100011
n(8)	000001	000001
n(9)	000101	001011
n(10)	011001	111001
n(11)	111101	110011
n(12)	110001	110001
n(13)	110101	011011
n(14)	001001	101001
n(15)	101101	000011
n(16)	100001	100001

Table 2: The power residue method, adapted to a 6 bit example with two possible multipliers. The algorithm is shown at (a), and the complete set of 16 pseudorandom output states is listed in this table at (b). Note the deviations from randomness apparent in the regular patterns seen in the two low order bits of each number.

treat the numbers as points in the interval 0 to 1. Some tests can handle either case, of which the Chi-square test is one. It can be applied directly to the digits produced, or to groupings of the digits, or we can divide the interval 0 to 1 into subintervals and see how many of the random numbers fall into each of the subintervals and apply the Chi-square test to see if the distribution is biased.

The Chi-square (symbol X^2) statistic looks somewhat formidable, but in reality is easy to work with. The formula is:

$$X^2 = \sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i}$$

$$= \frac{(o_1 - e_1)^2}{e_1} + \frac{(o_2 - e_2)^2}{e_2} + \ldots + \frac{(o_k - e_k)^2}{e_k}$$

where e_i is each expected frequency, and o_i is the actual observed frequency. If we had a generator which produced 250 digits we would expect each of the digits 0 through 9 to appear 25 times, although the digits might actually appear more or less than 25 times. Assume that your random number generator has just produced a series of 500 digits. You count all the zeros, ones, etc, and tabulate these observed frequencies (as in table 3) along with the expected frequency in each case of 50. You have counted 58 zeros, 28 ones, etc. You next put these observed and expected frequencies into the X^2 formula and arrive at an answer of 46.45 as shown. At this point we need to turn to a Chi-square distribution table, which can be found in the back of almost any statistics book. Table 4 shows a portion of such a table and will suffice for most of our needs for uses like this application of the Chi-square statistic. The table is entered after you calculate the "degrees of freedom" in the column labeled "v" and after you

Digit	0	1	2	3	4	5	6	7	8	9	Sum
Observed frequency	58	28	40	34	70	62	72	36	40	60	500
Expected frequency	50	50	50	50	50	50	50	50	50	50	500

$$X^2 = \frac{(58-50)^2}{50} + \frac{(28-50)^2}{50} + \frac{(40-50)^2}{50} + \ldots + \frac{(60-50)^2}{50} = \underline{\underline{46.56}}$$

Table 3: A random number sequence can be tested with various statistical measures. One excellent test is the Chi-square test, here illustrated with a hypothetical single digit decimal random number generator with the observed frequencies shown in a trial of 500 cycles of calculation. If the result were truly random, of course, the expected frequencies of each digit would be uniform. The Chi-square test involves calculating the characteristic number shown by the formula here (using this table's data). This characteristic number is then used with a statistical reference table of the Chi-square distribution and the number of degrees of freedom allowed by the statistics (here v=9), to check the quality of the pseudorandom sequence.



SCIENTIFIC SOFTWARE

DPFUN is a 16-digit precision scientific subroutine package written for Microsoft extended & disk BASIC interpreters, including TRS-80 Level II BASIC. The thirteen double-precision exponential, logarithmic, trigonometric & inverse trigonometric functions provide a valuable utility for serious engineering & scientific applications.

DPFUN uses truncated continued fraction algorithms that fully exploit the precision available in 64-bit binary floating point notation. The code has been optimized for speed & is designed to be easily appended to & called by any user program. The complete set of BASIC subroutines occupies 2.5K.

DPFUN source code, only $10 ppd.
DPFUN on 8-inch CP/M disk, $20 ppd.

Miken Optical Co.
53 Abbett Ave.
Morristown, NJ 07960
(201) 267-1210

Circle 234 on inquiry card.

TRS-80 disc software

WORD PROCESSOR Avoid work processor converted from other system. Hard to load object file & cannot store text in disk. Ours is specifically designed for the TRS-80. Written in BASIC. No special hardware and text limit. $39.

INVENTORY While others use inefficient sequential file, we use 9-digit key for fast on-line random access. Reports give order info, performance summary, etc. $39.

MAIL print report & labels sorted by any field. Random access, special buffering technique. $35

DATA BASE MANAGER You can maintain a data base & produce reports without any programming. Define fields, types, screen & report formats on-line. Almost use up the required 32K memory. $49.

KEY RANDOM ACCESS UTIL hashing, blocking, buffering, auto I/O error retry. Put your disk into optimal use. $19.

ACCOUNT manage client accounts & account receivable. Remark fields for general use. Automatic billing & transaction recording. 32K req. $59.

Full documented. Fast delivery. Our competitors offers $99 cassette word processor, $90 'on memory' inventory, inflexible mail system, 16K data base.

MICRO ARCHITECT
96 Dothan St.,
Arlington, MA 02174

Circle 214 on inquiry card.

save more than 20%!
NORTH STAR——XITAN

the smartest computers at the smartest price
DOUBLE DENSITY
HORIZON-1-16K kit, list $1599 .... **$1279**
Assembled & tested, list $1899 ...... **$1519**
HORIZON-2-32K kit, list $2249 .... **$1799**
Assembled & tested, list $2549 ...... **$2039**
PASCAL for NORTH STAR on Disk **$49**
Powerful NORTH STAR BASIC .... **FREE**
XITAN computers--NOW--With
QUAD DENSITY DISKS!
Famous XITAN Software & BASIC ..... **FREE**
HORIZON & XITAN are S-100, Z-80–THE BEST
INTERTUBE II Terminal, list $995 .... **$795**
NEW: Our VIDEO BOARD CODE on Disk! Now you can run our computers on a TV! Save $HUNDREDS
TV Code on DISK .... **$10** LISTING .... **FREE**
Business Software, Terminals, Printers, Computers in stock & special-ordered. Other brands at good discounts. Ask! Which Computers are best?
BROCHURE .... **FREE**

AMERICAN SQUARE COMPUTERS
KIVETT DR, JAMESTOWN NC 27282
(919) 883-1105

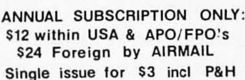
Circle 5 on inquiry card.

TRS-80 ATARI APPLE PET

THE RECREATIONAL PROGRAMMER™

QUALITY SOFTWARE!
TI & HP Programmables
BASIC for most computers

GAMES·ALGORITHMS·PUZZLES
STOCK ANALYSIS·AND MORE!
OVER 30 PAGES BIMONTHLY!







MAXIMUM VALUE FOR YOUR DOLLAR

NORTH STAR COMPUTER PRODUCTS

HORIZON1 16K KIT $1275.00
16K RAM BOARD KIT ...... $ 250.00
32K RAM BOARD KIT ...... $ 475.00

VERBATIM DISCS FOR NORTH STAR
BOX OF 10 .... $29. POST PAID

COMPLETE SYSTEMS AVAILABLE
CUSTOM SOFTWARE FOR NORTH STAR SYSTEMS

CASIO CALCULATORS AT DISCOUNT PRICES

MANY OTHER SUPER VALUES
WRITE OR CALL:

A.E.I.
3851 HACKETT AVE.
LONG BEACH, CALIF. 90808
(213) 421-4815 (213) 429-0535

Circle 21 on inquiry card.

ELECTROSENSITIVE PAPER

MANUFACTURED TO SPECS OF AXIOM, CENTRONICS, SCOPE, UNIVAC, RADIO SHACK, etc.

CALL:
301/840-9545

GENERAL SCIENTIFIC CORP
PO BOX 2145, MVS
GAITHERSBURG, MD 20760

VISA MASTER CHARGE

Circle 143 on inquiry card.

Scotch® BRAND **Diskettes**

Order Direct:
2 for $11.75
5 for $26.45
10 for $49.75 (Hard Box)
Price Includes Shipping
(Hawaii, Alaska & Canada add $1.00 shipping)
VISA, Master Charge, Check, Money Order

740-0 - 8" Soft Sector
740-32 - 8" Hard (32) Sector
744-0 - 5" Soft Sector
744-10 - 5" Hard (10) Sector
744-16 - 5" Hard (16) Sector

Mass. residents add 5% sales tax.

cpu COMPUTER PACKAGES UNLIMITED
99 Reservoir St. Holden, MA 01520
(617) 829-2570 Div. of SCB Inc.

Circle 67 on inquiry card.



Circle 380 on inquiry card.



Circle 358 on inquiry card.

v	$X^2_{.995}$	$X^2_{.99}$	$X^2_{.95}$	$X^2_{.90}$	$X^2_{.75}$
1	7.88	6.63	3.84	2.71	1.32
5	16.7	15.1	11.1	9.24	6.63
9	23.6	21.7	16.9	14.7	11.4
10	25.2	23.2	18.3	16.0	12.5
19	38.6	36.2	30.1	27.2	22.7
24	45.6	43.0	36.4	33.2	28.2

Table 4: An abbreviated portion of a standard Chi-square table used as described in the text to check the quality of a pseudorandom sequence.

determine the level of significance you want to test. The degrees of freedom in our case are $10 - 1 = 9$. This simply means that after we have determined nine of the observed frequencies, the 10th one is fixed. The frequencies have to add to 500, so we have "nine degrees of freedom." Traditionally, the Chi-square statistic is used to test the hypothesis that the numbers are randomly distributed. If the computed value of Chi-square is greater than the critical value read from the table, we would then conclude that the observed frequencies differ significantly from the expected frequencies and we would reject the hypothesis of randomness at whatever level of significance we select. The levels of significance often used are 0.05 and 0.01, corresponding to the $X^2_{.95}$ and $X^2_{.99}$ columns respectively, in the table. Going back to our example, we calculated a value of 45.56, but in the table for nine degrees of freedom and at the .01 level of significance, we see that the critical value of Chi-square is 21.7. Since $46.56 > 21.7$ we therefore conclude that the observed distribution of numbers produced by our generator differs significantly from the expected distribution at the 0.01 level of significance, and we therefore cast considerable suspicion on our random number generator. As previously mentioned, we could take our list of generated numbers in the 0 to 1 interval and set up some subdivisions of this interval. Next, we could see how many of the numbers fell into each subinterval, calculate the expected frequency for the subintervals, and apply the Chi-square test in the same fashion.

A second test frequently applied to random numbers is called the "poker test," but is in reality similar to the frequency test already considered. In the poker test we look for specific combinations of digits. For example, suppose we are generating 5 digit integer random numbers in the interval 00000 to 99999. Probability theory tells us the number of numbers we should have where all digits are the same, like 22222 or 66666 etc. We can also calculate the expected number of pairs, three-of-a-kind and full houses, etc. The Chi-square test can be applied to the analysis of the results.

A very similar test, called the "gap test," can be applied in like manner to the distances separating two like digits or two like groups of two or more digits. Again, Chi-square is a useful statistic in the analysis of these findings. The power residue method satisfactorily passes the poker test, the gap test and the usual frequency test; however it often fails to pass tests which consider runs of numbers. We've already seen how the power residue method produces certain predictable results, so this should not come as a surprise. However, if we are generating random *numbers* instead of random *digits*, this is not a big problem. A study of the runs up and down is often a good test to determine which multipliers are better than others when you use the power residue method, and the "run test" will also consistently prove that the Fibonacci series method will not produce the predicted number and lengths of runs. Taking, for example, a long string of random generator produced bits, we would count the number of strings of zeros bracketed by ones for each length, from one, on up to the longest string length. Number theory helps us determine the number of total runs we should have for both the ones and zeros. Figure 3 shows how to calculate these lengths, assuming "n" bits. There are several special tests similar to the run test, for example: "runs above and below the mean," etc. And, as usual, the Chi-square test is frequently applied to see if the actual results are reasonable.

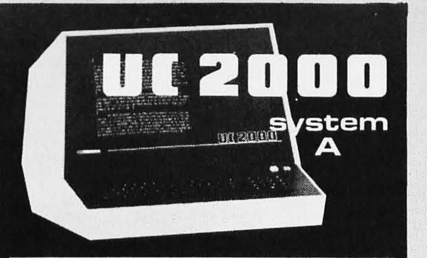
I can't guarantee all of the above will help you program your computer to play interesting Star Trek or sophisticated One-Armed-Bandit games but at least you'll be able to come up with generators that are biased in your favor.■

Figure 3: Several formulas for the run test of a pseudorandom sequence.

Run length	Formula
1	$(5n + 1) / 12$
2	$(11n - 14) / 60$
k (for $k < n - 1$)	$2 \left\{ (k^2 + 3k + 1)n - (k^3 + 3k^2 - k - 4) \right\} / (k + 3)!$
$n - 1$	$2 / n!$



Circle 383 on inquiry card.



The best choice in mainframes !

- S100 CARD FRAME
- 22 MHz 12" CRT MONITOR
- 18 AMP POWER SUPPLY
- UPPER & LOWER CASE ASCII KEY BOARDS
- AXIAL BLOWER
- ASSEMBLED & TESTED
- READY FOR YOUR CARDS
- $995.00

VERY ATTRACTIVE O.E.M. AND DEALER DISCOUNTS AVAILABLE.

INFINITE INCORPORATED

Celebrating Our 10th Year.

1924 WAVERLY PL., MELBOURNE, FL 32901 – PHONE (305) 724-1588

Circle 177 on inquiry card.

Custom Built Computer Desks

We custom build each computer desk to individual customer requirements. All surfaces are Formica; many wood grains & solid colors available. Send us your design, or let us design a module for you. Priced from $215. Call or write:

GOWAN INDUSTRIES
PO Box 470
Zoar, Ohio 44697
(216) 874-3666

Circle 154 on inquiry card.

APPLE II TINY BUSINESS SOFTWARE

ACCOUNTS RECEIVABLE — 100 customer accounts max, 8 transactions/month/customer (typical), month end and individual postings, automatic interest calculations. **$75.00**

ACCOUNTS PAYABLE — 100 payable accounts max, 8 transactions/monthly/payable (typical), month end posting. **$75.00**

PERPETUAL INVENTORY CONTROL — 100 suppliers, 900 individual inventory items, heuristic reorder procedure. **$100.00**

All of these programs have random record access, easy addition and deletion of items, password protection on sensitive procedures.

MIN. REQUIREMENT 48K APPLE II, one disk
OPT. REQUIREMENT 48K APPLE II, two disks

These procedures for screen environment, printer options available soon.

DOCUMENTATION only for the above $10 ea.

CHECK OR MONEY ORDER ONLY
DEALER INQUIRIES WELCOME

CUSTOM COMPUTING SYSTEMS INC.
204 - 2nd Ave. North, Saskatoon, Saskatchewan S7K 2B5 — (306) 242-7808

Circle 72 on inquiry card.



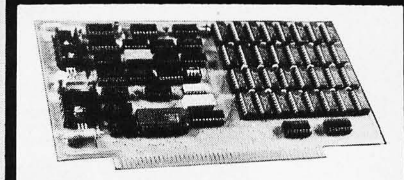
apple tv & computing

GRAND OPENING SALE!

An introductory sale! Every PET ordered this month will come with Full size ....
BIG KEYBOARD ....ONLY $895
or, the standard 8k PET $775
NEW!! ... PET MiniFloppy $595
16 K and 32 K PET NOW AVAILABLE
PET 2021 Printer.....................$549
80 column electrostatic with full graphics
PET 2022 Printer.....................$995
Forms tractor and graphics
Dual Drive Floppy for PET $1295
TRS 80 to S100 Interface (Kit).........$190
We Have The BALLY Computer System
And SOFTWARE for BALLY BASIC on audio cassettes. Write for list of titles.

apple tv & computing
(213) 559-4268
2606 S. Robertson Blvd.
Los Angeles, California 90034

Circle 10 on inquiry card.



64K RAM BOARD

The Z_S-SYSTEMS 64K RAM board is designed to operate in any Z80 based microcomputer having S-100 bus. It uses 16K dynamic RAM chips, & features:

- Board select
- Bank select
- Transparent on-board refresh
- 2 or 4MHz operation (w/ no wait state)
- Memory disable

Compatible with Cromemco system
Fully assembled, burned in, & tested
Available from stock to 60 days
As low as $500.00 in quantities of 100

Price of one $649.00
PC board only $59.00
With 16K RAM..................$359.00
Plus shipping charges

Z_S SYSTEMS
PO Box 1847, San Diego, CA 92112
(714) 447-3997

Circle 401 on inquiry card.

FOR TRS-80™ OWNERS T-PAL

Programming **A**mateur's **L**etter

THE "DO-IT-YOURSELF" SOFTWARE NEWSLETTER

Published Monthly, We'll Teach You All The Latest Wrinkles — How To Get The Most Out of Your Computer

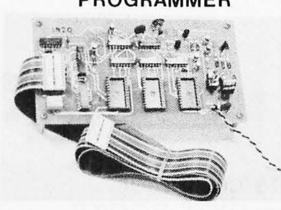
Graphics • Games • Personal Home • Business • Finance And Much More!

$24/year . . . or write for FREE Details

THE MAIL MART
Box B 11102 San Francisco, CA 94101

Circle 199 on inquiry card.

EPROM PROGRAMMER



The EDD 1001:
- Programs 2704, 2708, TMS2716, I2716, I2758
- Plugs into your empty EPROM socket
- Is provided with complete driver listing for 8080/Z80 systems (also available on EPROM)
- Requires external 30-37V unregulated

EDD 1001 EPROM Programmer Kit	$75 ppd
Driver program on EPROM	$25
Manual only (refundable with order) (CA residents add 6% sales tax)	$4

EPOCH DATA DEVICES
P.O. Box 22037
San Diego, CA 92122
(714) 481-1753

Circle 132 on inquiry card.

6800 SS50 16K SYSTEMS

Includes: Mainframe cabinet, mother board, power supply, fan, CPU, 16K static RAM, and choice of I/O card................................ **$1294.29**
Other packages available.

16K STATIC RAM BOARDS — SS50

Gold Bus connectors — DIP switch controllable addressing, write protect and enabeling of each 4K block. Tested at 2 MHz..... **Assembled $298.13**
Above but socketted and with software control registers. **Assembled $368.16**

TI TMS 4044's:
450 ns............................$5.00 each
250 ns............................$6.00 each
2708's............................$7.90 each

These are *factory prime* chips from same shipments we use in our professional quality boards.
Add $10 handling on orders under $200.

GIMIX INC.



1337 W. 37th Place • Chicago, IL 60609
(312) 927-5510 • TWX 910-221-4055

The Company that delivers.
Quality Electronic products since 1975.
GIMIX® and GHOST® are registered trademarks of GIMIX INC

Circle 144 on inquiry card.

Circle 36 on inquiry card.

What's New?

The Physicians Microcomputer Report

The Physicians Microcomputer Report is a monthly publication for doctors who wish to become better informed about the computer and its application in the field of medicine. Some of the features include: software news, calculator corner, computers in patient health care, microcomputer hardware news, the bargain market, and computer articles of special interest to the physician. Additionally, the report contains articles on nonmedical applications such as linking your computer to a stock portfolio information center. Another intent of this publication is to facilitate the exchange of information between physicians who own computers. For this purpose the magazine has a listing of user groups.

The Physicians Microcomputer Report is available for $25 a year and $12.50 for students. Contact Dr Gerald M Orosz, POB 6483, Lawrenceville NJ 08648.

Circle 591 on inquiry card.

Attention Readers, and Vendors. . .

Where Do New Products Items Come From?

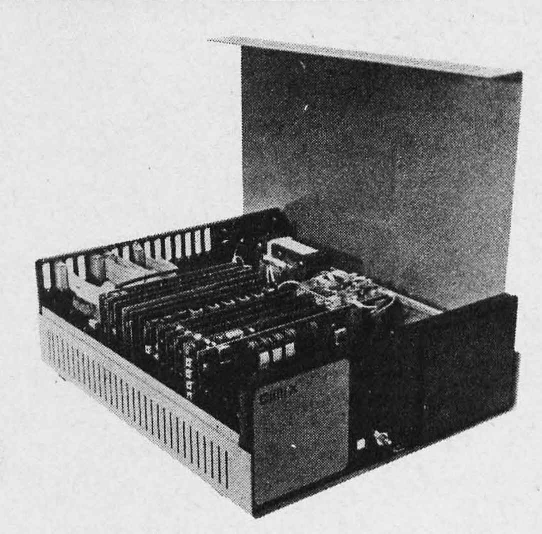
The information printed in the new products pages of BYTE is obtained from "new product" or "press release" copy sent by the promoters of new products. If in our judgment the information might be of interest to the personal computing experimenters and homebrewers who read BYTE, we print it in some form. We openly solicit releases and photos from manufacturers and suppliers to this marketplace. The information is printed more or less as a first in first out queue, subject to occasional priority modifications. While we would not knowingly print untrue or inaccurate data, or data from unreliable companies, our capacity to evaluate the products and companies appearing in the "What's New?" feature is necessarily limited. We therefore cannot be responsible for product quality or company performance.

Complete Microcomputer System from Gimix

Gimix Inc has announced its complete System 68 microcomputer. It features the following: a ferro-resonant constant voltage power supply; an SS-50 motherboard (15 50 pin and eight 30 pin gold plated slots); a 6800 processor board that holds four 2708s and three independently programmable software timers; the Gimix 16 K byte software readdressable static programmable memory boards organized into four separately controllable 4 K byte blocks, which allows the user as much memory as can be contained in the mainframe.

Dual-in-line package switch features allow use of existing SwTPC and MSI compatible software. The system is video based using the Gimix video board and advanced GMXBUG 3 K byte read only memory monitor that contains the standard utility functions plus routines that facilitate software development.

The price of $1395 includes the motherboard, switches, fan, power supply, video board, 3 K GMXBUG version 2, 8 K byte static programmable memory, 2 port parallel I/O (input/output) board, cable and two disk regulator board. For further information, contact Gimix Inc, 1337 W 37th Pl, Chicago IL 60609.



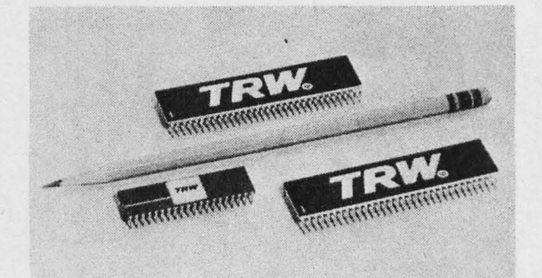
Circle 590 on inquiry card.

TRW LSI Products Introduces New Generation of Multipliers

A new series of monolithic multipliers, designated the MPY/HJ series, provide n by n bit multiplication of 24, 16, 12 and 8 bit numbers. All four multipliers have improved input registers that feature simplified clocking so that no data-hold time (clock overlap) is necessary. The three largest multipliers (MPY-24, -16 and -12HJ) feature improved output registers that can be made transparent for asynchronous output. They also feature a programmable selection of output product formats and can intermix two's complement numbers with numbers in absolute magnitude in the same operation.

All the new circuits are plug compatible with their first generation counterparts. Inserting one into a socket wired for an /AJ device automatically masks out the new /HJ features while still providing faster operation and reduced power consumption.

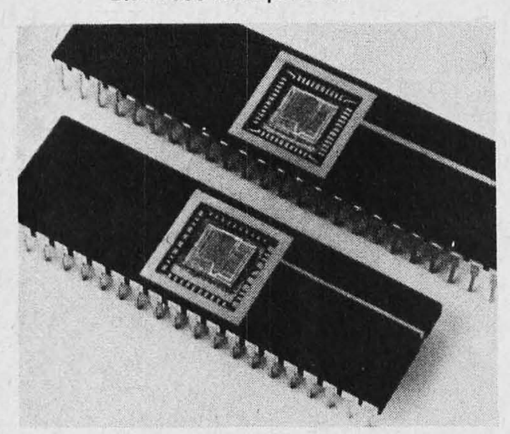
The MPY-24HJ contains a new shift and normalize feature, and yields a 48 bit product in 200 ns. The MPY-24HJ is supplied in a standard 64 pin dual-in-line package. The MPY-16HJ produces a 32 bit product in 100 ns and is pin compatible with the older 16 bit multipliers. Like the MPY-24HJ, it can be expanded to operate on 32, 48, 64 and larger numbers. The MPY-12HJ multiplies a pair of 12 bit numbers and yields their product in 80 ns. It is ideal for digital signal processing applications such as fast Fourier transforms and digital filters. The MPY-8HJ produces a 16 bit product in 65 ns. A fast version of the 8 bit multipliers called the MPY-8HJ-1 is being offered. It produces a 16 bit product in 45 ns and is intended for use in digital television systems.



Prices are $59 for the MPY-8HJ; $71 for the MPY-8HJ-1; $103 for the MPY-12HJ; $157 for the MPY-16HJ and $310 for the MPY-24HJ. For more information, contact TRW LSI Products, POB 1125, Redondo Beach CA 90278.

Circle 592 on inquiry card.

Zilog Announces Availability of 16 Bit Z8000 Microprocessor



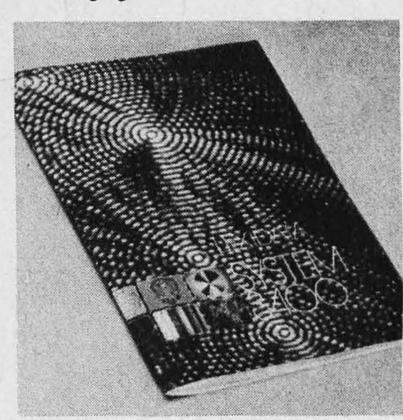
Zilog Inc has announced the availability of a Z8000 microcomputer processor circuit that offers users the architectural resources of mini and large mainframe computers in a single circuit device. The processor is available in two versions: the Z8001 in a 48 pin ceramic dual-in-line package that allows the user to address up to 8 M bytes of memory; and the Z8002 in a 40 pin ceramic dual-in-line package. The 40 pin Z8002, designed for smaller, less memory intensive applications, is compatible with the 48 pin Z8001, but the 40 pin processor's addressing is limited to 64 K bytes in each of its 6 address spaces.

A scaled N/MOS depletion load silicon gate device, the Z8000 processor densely packs 17,500 transistors on a device which is 238 by 256 mils. The Z8000 is designed for both minicomputer and microcomputer applications. The Z8000 contains 24, 16 bit registers that reduce the number of memory references needed in programming. Sixteen of those registers are general purpose. The Z8000's problem solving instruction set supports seven different data types from single to 32 bit words, has 8 addressing modes, and 418 usable op code combinations.

Pricing for the Z8001 is $195 for 1-9 pieces, $162.50 for 10-99 quantities and $140 for lots of 100 and up. The Z8002 sells for $150 for 1-9 quantities, $125 for 10-99 pieces, and $107.10 for lots of 100 and up. For further information contact Zilog, 10340 Bubb Rd, Cupertino CA 95014.

Circle 615 on inquiry card.

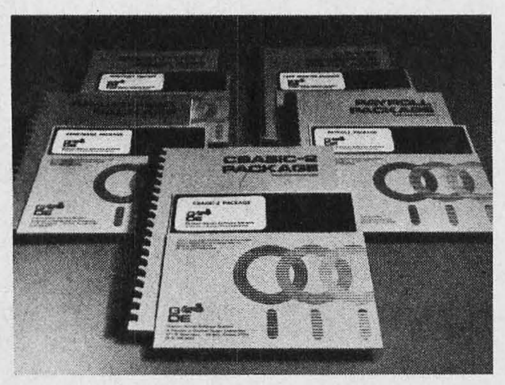
28 Page Brochure on Computer Graphics and Imaging



Lexidata Corporation, 215 Middlesex Turnpike, Burlington MA 01803 offers a free 28 page brochure describing the relative advantages of various display technologies and comparing them to its new System 3400 Video Image Processor. Interfacing, system software, application, and hardware options are covered in detail.

Circle 618 on inquiry card.

New Software Compatible With Any Z-80 or 8080 CP/M System



Circle 616 on inquiry card.

Graham-Dorian Software Systems has developed four complete software program packages for payroll, inventory, cash register, and apartment management. All programs are compatible with any Z-80 or 8080 CP/M system, and can be ordered in eight inch (double or single density) or five inch floppy disks. Each program package contains a disk with CBASIC-2 compiler, CBASIC-2 run command, the Graham-Dorian software program in INT and BAS file form, plus a users manual and hard copy source listing. The four programs sell for $695 each. One CBASIC-2 is free with a program order, others cost $89.95 each. For further information contact Graham-Dorian Software Systems, 211 N Broadway, Wichita KS 67202.

Text Processing Software

Digitan Systems Inc has announced a text processing system which uses special commands for text formatting applications. It is intended for use with 8080 and Z-80 microcomputer based systems. The commands include multiple line spacing; left and right margin control; indenting; paging; optional right margin justification; centering and underlining text; no-frill modes; automatic page numbering; page and line length control; and the printing of left, right and center header titles and footer titles with optionally different titles based on even and odd pages. Also included is the ability to input extra data from a file or the console terminal during the formatting process.

The text processing system will automatically loop for repeated formatting applications such as form letters. A preprocessing program is able to select a subset of the extra text data according to a user specified matching pattern. The output of the text processor can be directed to either the console terminal, line printer, or a disk file.

The source code of the text formatter has been written in CBASIC and runs under the CP/M operations system. It is available on eight inch floppy disks with a comprehensive manual at a cost of $250 per copy. A well documented source code is also available for an additional fee. For further information contact Digitan Systems Inc, 5001 16th Av, Brooklyn NY 11204.

Circle 617 on inquiry card.

A Powerful Disk Based Operating System for 6800 Microprocessors

The CP/68 operating system for the 6800 family of microprocessors furnishes big system features and capabilities for microcomputers. A combination of memory resident and transient commands provide the system's flexibility. With the CP/68 operating system it is possible to add your own commands to the system. PIP (Peripheral Interchange Program) allows transfer of data between physical devices. Wildcard operation of all disk commands lets the user specify files either ambiguously or unambiguously.

Other features of the operating system are complete device independent I/O (input/output); sequential and random file access methods; dynamic allocation and expansion of files; command files; and chaining and overlaying of user programs. It fits in less than 8 K bytes and can be relocated anywhere in memory; the extended instruction set includes 19 new 6809-type instructions (PSHX, PULX, etc); all disk operating system services are available through a single supervisor call; and it easily interfaces to new devices and peripherals.

The operating system supports functions that STRUBAL+ used to provide in its runtime package. The operating system runs on Percom, ICOM, MSI, Smoke Signal, Micropolis and SwTPC systems. For further information contact Hemenway Associates Inc, 101 Tremont St, Suite 208, Boston MA 02108.

Circle 619 on inquiry card.

Circle 4 on inquiry card.

S-100 Single and Double Density Disk System



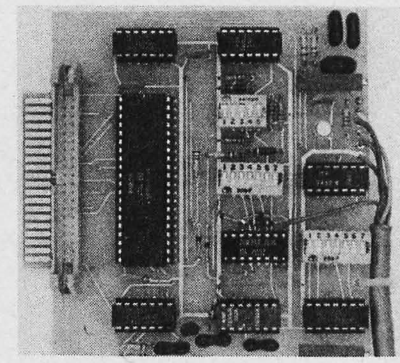
DISCUS 2D is a full-size, single and double density floppy disk system capable of storing up to 600 K bytes of data on each side of an 8 inch disk. This disk is formatted to be compatible with the IBM System 34. Like the original single density DISCUS I, DISCUS 2D comes fully assembled with a controller board and a Shugart SA800R full-size drive mounted in a cabinet with a power supply.

The S-100 controller board utilizes the Western Digital 1791 dual density controller device and also has power on jump circuitry, 1 K bytes of programmable memory, 1 K bytes of read only memory with built-in monitor, and a hardware universal asynchronous receiver-transmitter with a data rate generator to simplify I/O (input/output) interfacing. It is capable of handling up to four drives.

Software includes BASIC-V virtual disk BASIC, DOS and Disk/ATE assembler and editor. Extra cost optional software, including CP/M Microsoft Extended Disk BASIC and FORTRAN, is available. The price is $1149 for the completely assembled single and double density system, and $795 for each additional drive. For further information, contact Thinker Toys, 1201 10th St, Berkeley CA 94710.

Circle 585 on inquiry card.

TRS-80 Serial Input/Output Board



This board is RS-232 compatible and can be used with or without the expansion bus. There are on-board switch selectable data rates of 110, 150, 300, 600, 1200 and 2400 bps; parity odd, even, or null; 5 to 8 data bits and 1 or 2 stop bits. It has a data terminal ready line. The board alone sells for $19.95 (with parts $59.95). Assembled, it is $79.95. Contact Electronic Systems, POB 21638, San Jose CA 95151.

Circle 587 on inquiry card.

Light Pen Complements Apple II Computer

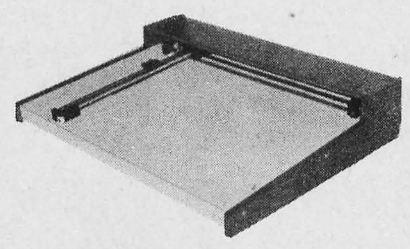


Symtec Inc, POB 462, Farmington MI 48024 has announced a low cost light pen for microcomputer use. The Symtec light pen is supplied complete with interface and provides an X, Y coordinate number to the bus when the pen is activated by a touch sensitive switch or from software control. The pen can provide up to 255 Y values and 511 X values and is software dividable to fit any screen size.

The Apple version of the light pen can resolve a single high resolution point and can be used with all of the Apple graphics features and text. This version is provided with a demonstration cassette written in integer BASIC for easy modification by the user if desired and allows use of the pen in the user's own programs. A complete listing of the light pen routine and suggested uses is included in the applications manual. The light pen is priced at $249.95.

Circle 586 on inquiry card.

X,Y Plotter Unit from Sylvanhills Laboratory



This X, Y plotter includes a plotter, drawing surface, electronics, and power supply completely assembled and ready for interface to any 8 bit transistor-transistor logic parallel port. The pen holder accepts any writing instrument or stylus 7 to 11 mm in diameter, encoded for 0.01 inches per pulse (0.005 inch optional). The maximum pen travel speed is 2.5 inches per second with a 24 V supply. A basic 8080 software program is included in the owner's manual. Applications include architectural, mechanical, and schematic drawing; printed circuit board artwork; positioning of small objects; computer generated art; games; and others.

The plot driver software is available as ASCII source files on paper tape and CP/M small disk formats. TEI and Cromemco small disk formats are also available. Both the BASIC and assembler source are provided, and contain comments which guide the user in making source modifications.

Unit-1 with an 11 by 17 inch drawing area is $1,049; Unit-2 with a 17 by 22 inch drawing area is $1,249. The plotters are also available in kit form with console and power supply priced separately. The owner's manual can be purchased for $5. For further information, contact Sylvanhills Laboratory Inc, POB 646, Pittsburg KS 66762.

Circle 588 on inquiry card.

TRS-80 Expandable Interface



Microtronix has introduced an expandable interface for the Radio Shack TRS-80. The basic interface unit uses low power Schottky circuitry, the standard Radio Shack 40 pin bus, and provides the following features: two joysticks for games, screen editing and educational instruction; stereo sound using two RCA 1863 programmable integrated circuits; parallel printer interface. At an introductory price of $129.95, the interface may be ordered with a $29.95 optional real time clock. Joysticks and music may be controlled directly from the user's BASIC program, using the INP and OUT commands. For further information, contact Microtronix, POB Q, Philadelphia PA 19105.

Circle 589 on inquiry card.

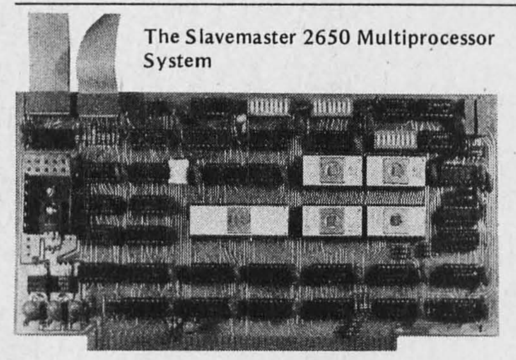
PERCOM DATA COMPANY, INC.
DEPT. B
211 N. KIRBY • GARLAND, TX. 75042

Computer Terminals Directory

A user-oriented directory of computer terminals has been published by the Association of Time-Sharing Users (ATSU). This guide contains photographs and full page information about each of 120 terminals. In addition to the latest pricing information, the directory lists each terminal's lease costs (when available), the number that have been installed, and information about whom to contact at each supplier. The *Computer Terminals Directory* is available for $45 in bound form. It is available as part of a membership in ATSU for $85, in loose leaf form, as it is part of the Association's three volume *Interactive Computing Directories.* Orders for the Directory or for Association membership should be sent to ATSU, POB 9003, Boulder CO 80301.

Circle 593 on inquiry card.

The Slavemaster 2650 Multiprocessor System



The Slavemaster 2650 S-100 bus multiprocessor system is based on the Signetics 2650 microprocessor. The system is composed of two identical S-100 cards interconnected by one ribbon cable. One is identified as the slave and the other the master. The only functional difference is that the master has the ability to reset, reset-jump, or stop the slave.

Both processors operate at full speed with fetch and execute cycles interleaved in such a way that precise single processor timing is maintained. Once synchronized, there is no interaction between the two processors. Communication between the two processors is through a common data base in the S-100 memory.

Some of the features of the Slavemaster card include Kansas City cassette interface, RS-232/20 mA serial I/O (input/output), keyboard interrupt on serial input, real time clock interrupt, power fail interrupt, eight vectored interrupts decoded on board, 4 K byte 2708 erasable read only memory sockets with dual-in-line package switches to select reset and power-on jump address. The kit is priced at $198 per board. For further information, contact Victoria Micro Digital, 401 Dundee St, Victoria TX 77901.

Circle 594 on inquiry card.

Western Digital System Speeds Up Pascal

A set of integrated circuits which directly executes the object code from a Pascal compiler has been developed by Western Digital Corp, 3128 Red Hill Av, POB 2180, Newport Beach CA 92663.

The 16 bit processor, which forms the basis of the Pascal Microengine product line, executes Pascal programs at least five times faster than conventional system software. The system uses the version of Pascal which was developed at the University of California at San Diego (UCSD). The UCSD Pascal software system includes a complete operating system with the Pascal compiler, BASIC compiler, file manager, screen-oriented editor, debugging program, and graphics package; all written in the Pascal language.

The four integrated circuits are the following LSI metal-oxide semiconductor components:

- an arithmetic device containing arithmetic and logic unit, microinstruction decoding, and the register file;
- a microsequencer device containing macroinstruction decoding, portions of the control circuitry, microinstruction counters, and I/O (input/output) control logic;
- two MICROM devices containing the microinstruction read only memories and microdiagnostics.

Direct execution of the p-code (pseudocode) produced by the Pascal compiler eliminates the previously required host operating system and p-code interpreter.

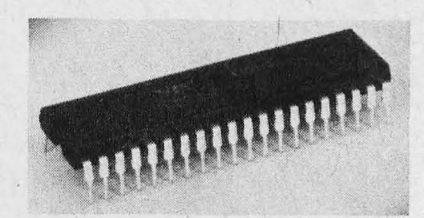
Additional features of the Microengine system include user-defined bus configuration, four levels of interrupts, single and multibyte instructions, hardware floating point operations, stack architecture, a 3.0 MHz 4 phase clock (75 ns per phase) and a transistor-transistor logic compatible three-state interface.

Circle 595 on inquiry card.

Stand-Alone Microprocessors

Three stand-alone microprocessors, the μPD8048, μPD8748, and μPD8035, have been announced by NEC Microcomputers Inc, 173 Worcester St, Wellesley MA 02181. The μPD8048 contains the following features normally found in external support devices: 1025 by 8 bits of read only memory; 64 by 8 bits of programmable data memory; 27 I/O (input/output) lines; 8 bit interval timer and event counter; and oscillator and clock circuitry.

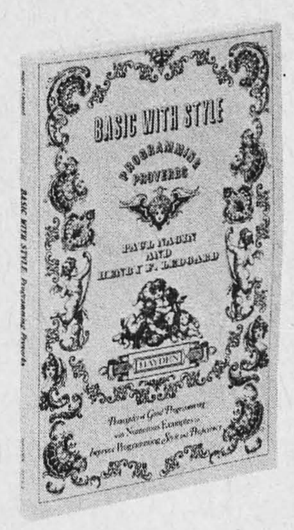
The μPD8748 (available late 1979) differs from the μPD8048 only in the use of an 1024 by 8 bit ultraviolet erasable read only memory for its program



memory, while the μPD8035 is scheduled for applications using external program memory. The functional power of the units can be expanded using standard 8080A/8085A peripherals and memory products. The microprocessors are available in a standard 40 pin, plastic or ceramic dual-in-line package.

Circle 597 on inquiry card.

BASIC With Style



BASIC With Style by Paul Nagin and Henry Ledgard is intended for BASIC programmers who want to write carefully constructed, readable programs. This 134 page book offers short rules and guidelines for writing more accurate, error free programs. These simple elements of style enable the programmer to focus creativity on the deeper issues in programming.

Chapter 1 is an overview. Chapter 2 is a collection of simple rules, called proverbs. The proverbs summarize the major ideas of the book in terse form. Chapter 3 is an introduction to a strict, top-down approach for programming problems in any programming language. The approach is oriented toward the easy writing of complete, correct, readable programs. Chapter 4 gives a set of strict program standards for writing programs, and Chapter 5 elaborates on several important and sometimes controversial ideas discussed in the chapter on programming proverbs.

The cost of the book is $5.95. It is available from Hayden Book Co Inc, 50 Essex St, Rochelle Park NJ 07662.

Circle 596 on inquiry card.



HOBBY WORLD®

CALL TOLL FREE: (800) 423-5387
CA, HI, AK: (213) 886-9200

TRS-80 FORTRAN PLUS $340

As advertized! Supplied on 2 mini-diskettes, and requires a 32K system with one disk drive. Written by Microsoft, designer of Level II and Level III. Includes Fortran compiler, Z-80 macro assembler, text editor, and linking loader. Adds speed and versatility.

BUY 7, GET 1 FREE!

**Buy 7 of one type, get the eighth of that type free!*

Order by type no.

1702A	$3.50
2708	10.00
4116	11.00
21L02-250	1.20
21L02-450	1.00
21L02-650	.65
2114-450	8.00
5203Q	3.25

CALIFORNIA COMPUTER SYSTEMS
XVI 16K STATIC RAM KIT $285

A true static ram board designed for the S-100 bus. Bank switching capability, addressable in 4K blocks. FR4 silk screened PC board with solder mask on both sides! The lowest price TRUE static ram board in its class.

- *IEEE S-100 Compatible.*
- *True static operation*
- *Requires only +5 volts*
- *450 ns*
- *Fully buffered*

Bareboard $27.00

ANADEX PRINTER Model DP-8000 $999

Connects easily to most popular computers including TRS-80. 3 basis ASCII compatible interface configurations are provided. 80 columns, 112 cps. 84 lines per min, bi directional printing. Out-of-paper detector, uses standard low-cost papers. 96 character set, 9x7 dot matrix characters. Original plus up to 3 copies. Skip-over-perforation control, double width printing, 8 programmable vertical tab positions. Excellent readability. Superior to other printers costing three times as much.

New! From California Computer Systems!
WIRE WRAP & SOLDERTAIL PROTOTYPING BOARDS
your choice $27

All S-100 signals labeled on-board. All circuitry uncommited except for 4 multiple regulator pads. High density hole configuration, over 3600 holes. On-board ground bus.

Double sided, plated thru FR4 PC board. Accepts 14, 16, 18, 24, 28, 40 pin IC's.

Cat No.	Wirewrap	$27
Cat No.	Soldertail	$27

COMPUCRUISE $165

Navigational computer for mobile or marine use! Features cruise control, fuel management, trip computer, timer/counter. Warns low fuel! Compensates for tire size, converts to metric! 44 functions. Mounts easily on dash. Easy to read bright blue digits. With installation instructions.
Cat No. 1166

TRS-80 LEVEL III BASIC $42

As advertised in March Interface. Loads on top of level II, turns your TRS-80 into a powerful system. Solves loading problems, cures keyboard "bounce". Software cassette, has the power of a hardware modification! Guaranteed satisfaction!
Cat No. 1332

DATA CASSETTES 10 for $17

Highest quality, leaderless! With protective plastic case.
Cat No. 1142

8" DISKS

- *Single density*
- *IBM Compatible*

$40 box of 10

Cat No.	Type
1145	32 sector holes, 1 index hole
1146	IBM 32, 3740, 3540, 3770, 3790

SHUGART SA-400 MINIFLOPPY DRIVE $295

Hard and soft sectoring, single density, 35 track. Requires power supply.
Cat No. 1154

TRS-80 ELECTRIC PENCIL $95

Character oriented word processing system. Produce mailing lists, business forms, large numbers of original correspondance, camera ready copy for printing...all on your TRS-80. No carriage returns or hyphenations, line formatting is done by the Electric Pencil! Also features right margin justifying, page numbering, and tilting, and many combinations of line length, page length, etc. For TRS-80 level 1 or 2, 16K, and virtually any printer.

California Computer Systems
MEMORY ADD-ON 16K $70

For APPLE TRS-80, EXIDY

Everything you need! Installs in minutes, no special tools, no soldering! 250 nsec.
Cat No. 1156

TRS-80 SOFTWARE CASSETTES

Cat No.	Description	Price
1093	SARGON CHESS, II.	19.95
1041	STAR TREK, II.	14.95
1036	SCI FI GAME SAMPLER, I/II.	5.95
1042	TAROT I/II	5.95
1179	CRIBBAGE I/II.	9.95
1192	REAL TIME LUNAR LANDER II.	7.95
1195	BRIDGE CHALLENGER II.	14.95
1186	AIR RAID I/II.	14.95
1187	PILOT I/II	14.95
1047	OTHELLO I/II	5.95
1043	SMALL BUSINESS BOOKKEEPING I/II.	14.95
1051	DAILY BIORHYTHM PROGRAM I/II.	5.95
1049	MICRO TEXT EDITOR I/II	9.95
1038	INVENTORY MODULAR I/II.	19.95
1153	EDIT-80, text editor II (32K)	39.95

S-100 COMPUTER BOARDS

CCS = California Computer Systems
WMC = Wameco
IA = Ithaca Audio
SSM = Solid State Music
SDS = SD Systems
SPL = Speechlab
HUH = HUH Electronics
MH = Mountain Hardware
a&t = assembled & tested

1601	CCS/M-XVI	16K STATIC RAM MODULE KIT	$285
1602		as above, a&t	
1603		as above, bareboard	
1500	HUH/S-100	MPA kit	
1503	MH	PROPROM, 8K EPROM BOARD	$214
1504		6834 EPROMS for above	$ 10
1505	MH	100,000 DAY CLOCK, a&t	$219
1506	MH	INTROL, 64 chan remote control a&t	$329
1507	WMC/QM1	12 SLOT MOTHERBOARD	$ 39
1508		as above, with connectors	$ 80
1509	WMC/MEM1	8K STATIC RAM BOARD	$ 28
1510		parts only for above	$ 80
1403	SSM/CB1	8080A CPU BOARD KIT	$134
1408	SSM/SB1	MUSIC SYNTHESIZER KIT	$ 145
1411	SSM/IO4	2 PARALLEL + 2 SERIAL PORTS KITS	$139
1414	SSM/IO2	IO UNIVERSAL BOARD KIT	$ 48
1417	SSM/VB1B	VIDEO INTERFACE KIT	$129
1425	SSM/MB3	2/4K EPROM BOARD KIT	$ 54
1420	SSM/MB4	2 MHZ STATIC RAM KIT	$ 89
1427	SSM	ALTAIR IMSAI EXTENDER BOARD	$ 10
1428		connector for above	$ 4
1429	SSM/OB1	VECTOR JUMP & PROTO-TYPING CARD KIT	$ 47
1432	SSM/MT1	15 SLOT MOTHERBOARD	$ 39
1433	SSM/MB8A	16K (2708) EPROM BOARD KIT	$ 88
1436	SSM/MB9	4K STATIC PROM/RAM BOARD KIT	$ 64
1438	SSM/VB2	VIDEO BOARD KIT	$139
1511	IA	2708/2716 EPROM BARE-BOARD	$ 28
1512	IA	Z80 CPU BAREBOARD	$ 32
1513	IA	8K STATIC RAM BARE-BOARD	$ 28
1514	IA	S-100 WIREWRAP BOARD	$ 28
1600	CCS	S-100 WIREWRAP BOARD	$ 29
1516	SDS	VERSAFLOPPY KIT	$159
1517-0	SDS	EXPANDORAM KIT	$185
1517-16		as above, with 16K RAM	$249
1517-32		as above, with 32K RAM	$330
1517-48		as above, with 48K RAM	$425
1517-64		as above, with 64K RAM	$500
1165	NEWTECH	MUSIC BOARD a&t	$ 57
1518	SPL	32 WORD SPEECHLAB a&t	$189
1520	SPL	64 WORD SPEECHLAB a&t	$299
1222	uSOUNDER	SOUND EFFECTS BOARD a&t	$149

VERBATIM 5¼" DISKETTES
$27 box of 10

Cat No.	Type	Use
1147	Soft sector	TRS-80, Apple
1148	Hard, 10 hole	North Star
1149	Hard, 16 hole	Micrppolis

SEND FOR FREE SPRING CATALOG FEATURING:

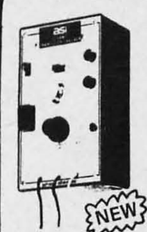
FACTORY FRESH, PRIME IC'S, LED'S, READOUTS, RECTIFIERS, ZENERS, TRS-80 ADD-ONS, S-100 BOARDS, PC AIDS, TEST EQUIPMENT, BOOKS, SOFTWARE, AND MORE!

Pay by check, COD, Visa, or Mastercharge. Order by phone or mail. Please include phone no. USA add $1.50 for shipping/handling, or $2.50 for air. Foreign add $2.00 for surface, $5.00 for air. COD's add 85c. All items guaranteed satisfaction for 120 days!

19355 BUSINESS CTR DR 6B6 NORTHRIDGE, CA 91324

Circle 170 on inquiry card.

Transistor Checker



— Completely Assembled —
— Battery Operated —

The ASI Transistor Checker is capable of checking a wide range of transistor types, either "in circuit" or out of circuit. To operate, simply plug the transistor to be checked into the front panel socket, or connect it with the alligator clip test leads provided. The unit safely and automatically identifies low, medium and high-power PNP and NPN transistors. Size: 3¾" x 6¼" x 2"
"C" cell battery not included.

NEW

Trans-Check $29.95 ea.

Custom Cables & Jumpers

NEW NEW

DB 25 Series Cables

Part No.	Cable Length	Connectors	Price
DB25P-4-P	4 Ft.	2-DP25P	$15.95 ea.
DB25P-4-S	4 Ft.	1-DP25P/1-25S	$16.95 ea.
DB25S-4-S	4 ft.	2-DP25S	$17.95 ea.

Dip Jumpers

DJ14-1	1 ft.	1-14 Pin	$1.59 ea.
DJ16-1	1 ft.	1-16 Pin	1.79 ea.
DJ24-1	1 ft.	1-24 Pin	2.79 ea.
DJ14-1-14	1 ft.	2-14 Pin	2.79 ea.
DJ16-1-16	1 ft.	2-16 Pin	3.19 ea.
DJ24-1-24	1 ft.	2-24 Pin	4.95 ea.

For Custom Cables & Jumpers, See JAMECO 1979 Catalog for Pricing

CONNECTORS
25 Pin-D Subminiature

DB25P (as pictured)	PLUG (Meets RS232)		$2.95
DB25S	SOCKET (Meets RS232)		$3.50
DB51226-1	Cable Cover for DB25P or DB25S		$1.75

PRINTED CIRCUIT EDGE-CARD

.156 Spacing-Tin-Double Read-Out — Bifuracted Contacts — Fits .054 to .070 P.C. Cards

15/30	PINS (Solder Eyelet)		$1.95
18/36	PINS (Solder Eyelet)		$2.49
22/44	PINS (Solder Eyelet)		$2.95
50/100 (.100 Spacing)	PINS (Wire Wrap)		$6.95
50/100 (.125 Spacing)	PINS (Wire Wrap)	R681-1	$6.95



Solar Cells 2x2cm

- 0.4 volts
- 100mA
- 41 MW

Can be added in series for higher voltage or parallel for higher current.

#SC 2x2 $1.95 ea. or 3/$5.00

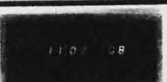
the 3rd Hand

MAKES CIRCUIT ASSEMBLY A BREEZE!

Lets you work with both hands. Sturdy Aluminum Construction. **$9.95 ea.**



- **Clamp "3rd Hand" on edge of bench, table or workboard. Insert circuit board, position components.**
- **Flip circuit board to flat position for soldering and clipping.**



- Bright .300 ht. comm. cathode display
- Uses MM5314 clock chip
- Switches for hours, minutes and hold modes
- Hrs. easily viewable to 20 ft.
- Simulated walnut case
- 115 VAC operation
- 12 or 24 hr. operation
- Incl. all components, case & wall transformer
- Size: 6¾" x 3-1/8" x 1¾"

JE701

6-Digit Clock Kit $19.95

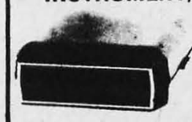
REMOTE CONTROL TRANSMITTER & RECEIVER



- CAN BE USED AS REMOTE CONTROL FOR TV
- USE IT FOR YOUR OWN GARAGE DOOR OPENER
- TRANSMITS BETWEEN 38KHz AND 41.5KHz
- THOUSANDS OF USES FOR REMOTE CONTROL APPLICATIONS
- TRANSMITTER USES A 1.5V BATTERY
- SCHEMATIC INCLUDED

$19.95

INSTRUMENT/CLOCK CASE



This case is an injection molded unit that is ideal for uses such as DVM, COUNTER, or CLOCK cases. It has dimensions of 4½" in length by 4" in width by 1-9/16" in height. It comes complete with a red bezel.

PART NO: IN-CC $3.49 each

MICROPROCESSOR COMPONENTS

8080A/8080A SUPPORT DEVICES

8080A	CPU	$ 9.95
8212	8-Bit Input/Output	3.25
8214	Priority Interrupt Control	5.95
8216	Bi-Directional Bus Driver	3.49
8224	Clock Generator/Driver	3.95
8226	Bus Driver	3.49
8228	System Controller/Bus Driver	5.95
8238	System Controller	5.95
8251	Prog. Comm. 1/0 (USART)	7.95
8253	Prog. Interval Timer	14.95
8255	Prog. Periph. 1/0 (PPI)	9.95
8257	Prog. DMA Control	19.95
8259	Prog. Interrupt Control	19.95

6800/6800 SUPPORT DEVICES

MC6800	MPU	$14.95
MC6802CP	MPU with Clock and Ram	24.95
MC6810API	128X8 Static Ram	5.95
MC6821	Periph. Inter. Adapt (MC6820)	7.49
MC6828	Priority Interrupt Controller	12.95
MC6830L8	1024X8 Bit ROM (MC68A30-8)	14.95
MC6850	Asynchronous Comm. Adapter	7.95
MC6852	Synchronous Serial Data Adapt.	9.95
MC6860	0-600 bps Digital MODEM	12.95
MC6862	2400 bps Modulator,	14.95
MC6880A	Quad 3-State Bus. Trans. (MC8T26)	2.25

MICROPROCESSOR CHIPS—MISCELLANEOUS

Z80(780C)	CPU	$19.95
Z80A(780-1)	CPU	24.95
CDP1802	CPU	19.95
2650	MPU	19.95
8035	8-Bit MPU w/clock, RAM, 1/0 lines	19.95
P8085	CPU	19.95
TMS9900JL	16-Bit MPU w/hardware, multiply & divide	49.95

SHIFT REGISTERS

MM500H	Dual 25 Bit Dynamic	$.50
MM503H	Dual 50 Bit Dynamic	.50
MM504H	Dual 16 Bit Static	.50
MM506H	Dual 100 Bit Static	.50
MM510H	Dual 64 Bit Accumulator	.50
MM5016H	500/512 Bit Dynamic	.89
2504T	1024 Dynamic	3.95
2518	Hex 32 Bit Static	4.95
2522	Dual 132 Bit Static	2.95
2524	512 Static	.99
2525	1024 Dynamic	2.95
2527	Dual 256 Bit Static	2.95
2528	Dual 250 Static	4.00
2529	Dual 240 Bit Static	4.00
2532	Quad 80 Bit Static	2.95
2533	1024 Static	2.95
3341	Fifo	6.95
74LS670	4X4 Register File (TriState)	2.49

UART'S

A-Y-5-1013	30K BAUD	5.95

MICROPROCESSOR MANUALS

M-Z80	User Manual	$7.50
M-CDP1802	User Manual	7.50
M-2650	User Manual	5.00

ROM'S

2513(2140)	Character Generator(upper case)	$9.95
2513(3021)	Character Generator(lower case)	9.95
2516	Character Generator	10.95
MM5230N	2048-Bit Read Only Memory	1.95

RAM'S

1101	256X1	Static	$1.49
1103	1024X1	Dynamic	.99
2101(8101)	256X4	Static	3.95
2102	1024X1	Static	1.75
21L02	1024X1	Static	1.95
2111(8111)	256X4	Static	3.95
2112	256X4	Static MOS	4.95
2114	1024X4	Static 450ns	9.95
2114L	1024X4	Static 450ns low power	10.95
2114-3	1024X4	Static 300ns	10.95
2114L-3	1024X4	Static 300ns low power	11.95
5101	256X4	Static	7.95
5280/2107	4096X1	Dynamic	4.95
7489	16X4	Static	1.75
74S200	256X1	Static Tristate	4.95
93421	256X1	Static	2.95
UPD414 (MK4027)	4K	Dynamic 16 pin	4.95
UPD416 (MK4116)	16K	Dynamic 16 pin	14.95
TMS4044-45NL	4K	Static	14.95
TMS4045	1024X4	Static	14.95
2117	16,384X1	Dynamic 350ns (house marked)	9.95
MM5262	2KX1	Dynamic	4/1.00

PROM'S

1702A	2048	FAMOS	$5.95
TMS2516 (2716)	16K*	EPROM(Intel 2716)	49.95
	*Requires single +5V power supply		
TMS2532	4KX8	EPROM	89.95
2708	8K	EPROM	10.95
2716 T.I	16K**	EPROM	29.95
	**Requires 3 voltages, —5V, +5V, +12V		
5203	2048	FAMOS	14.95
6301-1(7611)	1024	Tristate Bipolar	3.49
6330-1(7602)	256	Open C Bipolar	2.95
82S23	32X8	Open Collector	3.95
82S115	4096	Bipolar	19.95
82S123	32X8	Tristate	3.95
74186	512	TTL Open Collector	9.95
74188	256	TTL Open Collector	3.95
74S287	1024	Static	2.95

CSC CONTINENTAL SPECIALTIES
PROTO BOARDS

Proto Board 203



A total ready-to-use power breadboard prototest device with a built-in regulated, short-proof power supply. Just plug-in and start building! 2 extra floating 5-way binding posts for external signals. Self-contained with power switch indicator lamp and power fuse. 24-14 pin DIP capacity. Attractive two-tone quality case. All metal construction. 9⅝"L x 6½"W x 2⅝"H 5 lbs.

PB 203 **$75.00**

Proto Board 203A

All the features of the PB-203 plus additional power supply flexibility, provides 1%-Regulated 5VDC supply (same specifications as PB-203). Regulated separate +15VDC and −15 VDC 0.5A supplies, each with internally and independently adjustable output voltage. Ripple and noise of + and −15V supplies. 10mV at 0.25A.

PB 203A **$124.95**

Model Number	L x W x H (Inches)	Price
PB-6	6.0 x 4.5 x 1.4	$15.95
PB-100	6.0 x 4.5 x 1.4	$19.95
PB-101	6.0 x 4.5 x 1.4	$22.95

Model Number	L x W x H (Inches)	Price
PB-102	7.0 x 4.5 x 1.4	$26.95
PB-103	9.0 x 6.0 x 1.4	$44.95
PB-104	9.8 x 8.0 x 1.4	$54.95

B+K PRECISION 3½-Digit Portable DMM



- Overload Protected
- 3" high LED Display
- Battery or AC operation
- Auto Zeroing
- 1mv, 1Va, 0.1 ohm resolution
- Overange reading
- 10 meg input impendence
- DC Accuracy 1% typical

Ranges: DC Voltage 0-1000V
AC Voltage 0-1000V
Freq Response 50-400 HZ
DC/AC Current: 0-100mA
Resistance 0-10 meg ohm
Size 6.4" x 4.4" x 2"

Model 2800 $99.95

Comes with test leads, operating manual and spare fuse

Accessories:

AC Adapter BC-28	$9.00
Rechargeable Batteries BP-26	20.00
Carrying Case LC-28	7.50

CSC 100 MHz 8-Digit Counter

CONTINENTAL SPECIALTIES CORPORATION

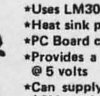
- 20 Hz-100 MHz Range
- .6" LED Display
- Crystal-controlled timebase
- Fully Automatic
- Portable — completely self-contained
- Size — 1.75" x 7.38" x 5.63"
- Four power souces, i.e. batteries, 110 or 220V with charger 12V with auto lighter adapter and external 7.2-10V power supply

MAX-100 $134.95

ACCESSORIES FOR MAX 100:
Mobile Charger Eliminator use power from car battery — **Model 100 — CLA $3.95**
Charger/Eliminator use 110 V AC — **Model 100 — CAI $9.95**

REGULATED POWER SUPPLY
NEW

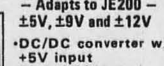
JE200
5V-1 AMP POWER SUPPLY



- Uses LM309K
- Heat sink provided
- PC Board construction
- Provides a solid 1 amp @ 5 volts
- Can supply up to ±5V, ±9V and ±12V with JE205 Adapter
- Includes components, hardware & instructions
- Size: 3½"x5"x2"H

JE200 $14.95

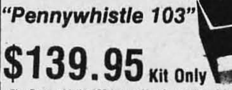
JE205
ADAPTER BOARD
— Adapts to JE200 — ±5V, ±9V and ±12V



- DC/DC converter w/ +5V input
- Toriodal hi-speed switching XMFR
- Short circ. protection
- PC Brd. construction
- Piggy-back to JE200 board
- Size: 3½"x2"x9/16"h

JE205 $12.95

The Incredible "Pennywhistle 103"



$139.95 Kit Only

The Pennywhistle 103 is capable of recording data to and from audio tape without critical speed requirements for the recorder and it is able to communicate directly with another modem and terminal for telephone "hamming" and communications. In addition, it is free of critical adjustments and is built with non-precision, readily available parts.

Data Transmission Method .....Frequency-Shift Keying, full-duplex (half-duplex selectable).
Maximum Data Rate..........300 Baud.
Data Format.................Asynchronous Serial (return to mark level required between each character).
Receive Channel Frequencies...2025 Hz for space; 2225 Hz for mark.
Transmit Channel Frequencies..Switch selectable: Low (normal) = 1070 space, 1270 mark; High = 025 space, 2225 mark.
Receive Sensitivity............−46 dbm accoustically coupled.
Transmit Level...............−15 dbm nominal. Adjustable from −6 dbm to −20 dbm.
Receive Frequency Tolerance...Frequency reference automatically adjusts to allow for operation between 1800 Hz and 2400 Hz.
Digital Data Interface..........EIA RS-232C or 20 mA current loop (receiver is optoisolated and non-polar).
Power Requirements..........120 VAC, single phase, 10 Watts.
Physical......................All components mount on a single 5" by 9" printed circuit board. All components included.

Requires a VOM, Audio Oscillator, Frequency Counter and/or Oscillosoope to align.

TRS-80 16K Conversion Kit

Expand your 4K TRS-80 System to 16K. Kit comes complete with:

- 8 each UPD416-1 (16K Dynamic Rams) 250NS
- Documentation for conversion

TRS-16K $99.95

COMPUTER CASSETTES



- 6 EACH 15 MINUTE HIGH QUALITY C-15 CASSETTES
- PLASTIC CASE INCLUDED 12 CASSETTE CAPACITY
- ADDITIONAL CASSETTES AVAILABLE #C-15-$2.50 ea

CAS-6 $14.95
(Case and 6 Cassettes)

SUP 'R' MOD II
UHF Channel 33 TV Interface Unit Kit



Wide Band B/W or Color System

- Converts TV to Video Display for home computers, CCTV camera, Apple II, works with Cromeco Dazzler, SOL-20, IRS-80, Challenger, etc.

MOD II is pretuned to Channel 33 (UHF).

- Includes coaxial cable and antenna transformer.

MOD II $29.95 Kit

SLIT-N-WRAP

Model P180 includes 2-100' spools #28 AWG wire wrap wire

Supplies insulated wire from spool to wrap-posts without prestripping and precutting using "daisy chain" method.

Model P180 Tool $24.50

P180A Replacement Bit	**$12.95**
W28-2 Replacement wire (3 spool pkg.) ..	**$2.75 ea.**

Specify color: A-green B-red C-clear D-Blue

IDEAL FOR TRS 80 — CASSETTE CONTROLLER



"Plug/Jack interface to any computer system requiring remote control of cassette functions"

The CC100 controls cassette motor functions, monitors tape location with its internal speaker, and requires no power. Eliminates the plugging and unplugging of cables during computer loading operation from cassette.

#CC-100 $29.50

63-Key Unencoded Keyboard



This is a 63-key, terminal keyboard newly manufactured by a large computer manufacturer. It is unencoded with SPST keys, unattached to any kind of PC board. A very solid molded plastic 13 x 4" base suits most application. IN STOCK **$29.95/each**

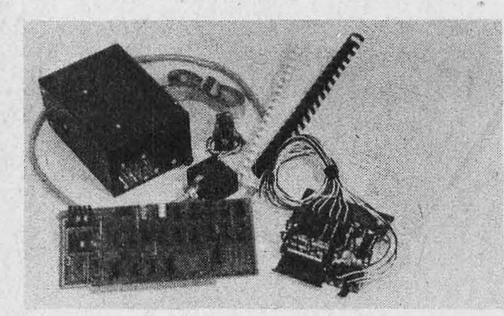
Hexadecimal Unencoded Keypad



19-key pad includes 1-10 keys, ABCDEF and 2 optional keys and a shift key. **$10.95/each**

Circle 200 on inquiry card.

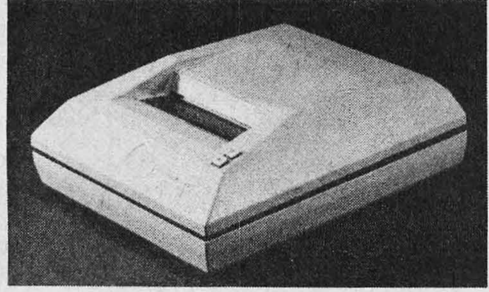
Universal Interface Converts IBM Selectric Typewriters



This universal interface unit has been designed for IBM Selectric typewriter conversions. The unit will interface to any RS-232, IEEE-488 or parallel port. A microprocessor is included on the circuit board for data flow control, formatting and character set selection. Installation on the Selectric is easy and does not affect normal typewriter operation. For those who do not want to convert their own typewriter, the company provides factory installation service. Selectric typewriters with conversion systems installed in accordance with factory instructions are still eligible for IBM warranty and service provisions. For further information, contact ESCON Products Inc, 171 Mayhew Way, Suite 204, Pleasant Hill CA 94596.

Circle 581 on inquiry card.

Printerm Model 879 Micro/Mini Printer



The Model 879 Micro/Mini printer is a high speed bidirectional printer which prints 120 characters per second at 75 lines per minute. It has a 9 by 7 or 9 by 9 high density matrix format, and provides up to four copies. The Model 879 has an ASCII 96 character set (upper case, lower case and triple wide expanded) and is operator switch selectable for an 80 or 132 column format. This RS-232 and parallel interface printer is available with roll paper feed, combination pin form and roll feed, or tractor feed. It contains 2 K bytes of memory for full page video dump. The price is $1395 for the standard model. For further information, contact Printer Terminals Corp, POB 535, Ramona CA 92065.

Circle 582 on inquiry card.

Low Cost Modification to DECwriter Printer Adds Graphics

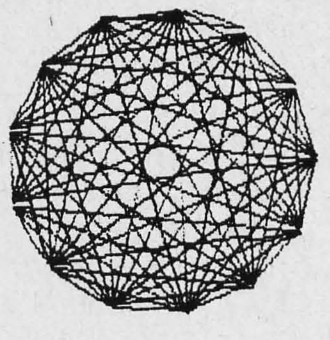
The Graphics II system is a low cost graphics modification for the Digital Equipment DECwriter II printer. It is available to upgrade existing printers, or can be factory installed with a new printer. The Graphics II system consists of a replacement circuit board for the DECwriter II and is plug compatible with internal cables for simple installation. The new circuit board uses the Fairchild F8 microprocessor.

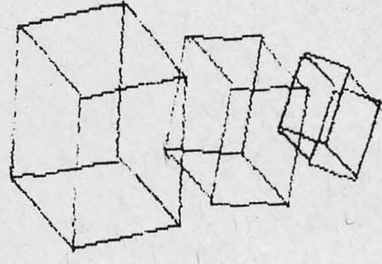
The DECwriter is a dot-matrix printer, and the Graphics II allows printing of a dot anywhere on the page. Bidirectional line feed is introduced, and the Vector Graphics capability allows the printing of a line between any two points on the page by using ASCII characters to specify the end point coordinates. This means that graphics can be generated by using the printer keyboard.

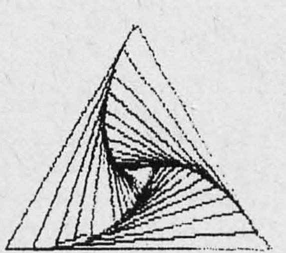
ASCII and APL character sets are standard, and other character sets can be used. Characters can be printed in any of four rotational orientations, and printed normal size, heavy bold face, or expanded width. The printing of bar code is also available.

The average printer speed has been increased to 50 characters per second, and data may be transmitted in bursts of up to 1000 characters at 1200 bps. Other features included as standard are EIA RS-232, 20 mA current loop and TTL interfaces, auto linefeed, top of form, and horizontal and vertical tabs.

The Graphics II system is priced at







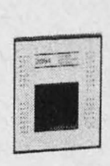
$850 as a field installed circuit card. For further information, contact Selanar Corp, 3054 Lawrence Expressway, Santa Clara CA 95051.

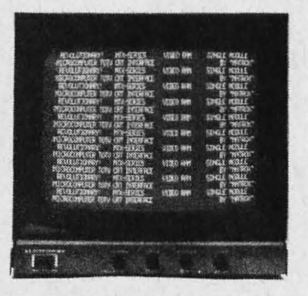
Circle 584 on inquiry card.

High Density Video Programmable Memory Module

The MTX-2064 and MMD-2480 are new members of the Matrox video programmable memory family (VRAM) of TV video controllers. The family provides an interface between any microprocessor and a TV monitor. On the input side the VRAMs look like a 1280 or 4098 by 8 bit static programmable memory with an access time of 500 ns. The output of the MTX-2064 is a video signal providing a flicker free display of 20 lines by 64 upper and lower case characters. The MMD-2480 provides 24 lines of 80 characters and limited graphics capability. No external refresh or memory is required. Any character may be displayed normally, inversely or blinking.

The bus structure permits direct interfacing to most micro and minicomputers. Being part of memory, the full power of the processor instruction set is available for display manipulations. A universal phase lock loop module permits the MMD-2480 to be locked to an external sync source such as a TV camera. Another option





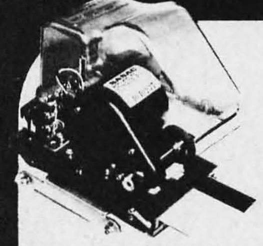
available for both models is a choice of American and European television standard field rates.

The units are completely self-contained and ready to use, including sync generator, programmable memory, read only memory and bus interface. They are housed in pin compatible 4.5 by 6 by 0.5 inch (11.43 by 15.24 by 1.27 cm) modules and draw under 800 mA from a single +5 V power supply. The MTX-2064 is priced at $295 and the MMD-2480 is $395. For further information, contact Matrox Electronic Systems Ltd, 2795 Bates Rd, Montreal, Quebec CANADA H3S 1B5.

Circle 583 on inquiry card.

California Digital

Post Office Box 3097 B • Torrance, California 90503



Sankyo Magnetic Card Reader

$59

These Sankyo I/O units are capable of storing and retrieving over 400 characters of data in under two seconds.
The flexability of this device lends itself to numerous applications. As an input reader to a computerized security system, the computer has the ability of identifying the card holder and admitting only those individuals who are authorized to enter the premises during specified time frames. The device is also suitable for maintaining customer information files, or any other application where small amounts of information must be quickly entered into a data processing system.
Accepts 2" by 4" HP style mag-cards. (Similar to bank cards.)
Motorized feeder pulls the magnetic card across the four channel read/write head. NEW surplus, original cost $200. Full documentation

S-100 Mother Board

Quiet Buss™

$29.95
8803-18
18 slot
IMSAI

The Quiet Buss from California Industrial is quality engineered. No short cuts have been taken to produce this mother board. Active termination circuitry prevents noise and crosstalk. Manufactured from extra heavy FR-4 epoxy glass

TELETYPE® MODEL 43

Even if we have to give them away, we're going to ship more 43's in 1979 than the aggregate of all our competitors.

Model 43AAA TTL)

EACH	3	10	25
$925.	875.	850.	825.

RS-232 Interface "K" Add $75.00 plus shipping

HEXADECIMAL KEYBOARD

$34.95



Maxi-Switch hexadecimal keyboards are designed for microcomputer systems that require 4-bit output in standard hex code.

Each assembly consists of 16 hermetically sealed reed switches and TTL "one shot" debounce circuitry.

Reliable low friction acetal resin plungers are credited for the smooth operation and long life of this premium keyboard.

Requires single +5 volt supply.

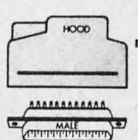
$29.95 DISKETTES

BOX of 10

Verbatim. APPLE/TRS-80 Mini - Soft sector

CONNECTORS





your choice
DB25P male plug & hood
or
DB25S female

$3.95

Qty.	fe.	male	hd.
10	3.45	2.45	1.15
25	3.15	2.25	1.05
100	2.85	1.90	.95
500	2.25	1.60	.85
1K	1.97	1.37	.73

Edge Connectors



GOLD
100 PIN
IMSAI/ALTAIR

Imsai solder .125x.250	$3.95	3/$ 9.00
Imsai w/w .125centers	$4.95	3/$13.00
Altair soldertail .140 row	$5.95	3/$15.00
SPECIALS		
22/44 Kim eyelet .156"	$1.95	3/$5.00
25/50 solder tab .156"	$1.09	3/$2.00
36/72 wide post w/w .156	$1.95	3/$5.00

SPERRY UNIVAC KEYBOARD

$24.88

The famous Sperry Univac 1710 Hollerith keyboard assembly is now available from California Industrial for only $24.88. The ideal computer input device for accountants and mathematicians. The numeric keys are placed on the lower three rows to resemble a ten key adding machine. This format allows one handed numeric data entry.
Original cost was $385. Used but guaranteed in excellent condition. Complete with documentation.

Apple Owners: TEN KEY

Data Entry Pad

$79.50

Plugs directly into you Apple II. Allows you to enter numerics, punctuation and upper case alpha characters, all from the data entry pad. Sold assembled in walnut finished enclosure.

SPECiAL

APPLE II
16K MEMORY
COLOR • GRAPHICS • SOUND

$1024 Mfg. Sug. Retail.... $1195

PLUS SHIPPING

Scotch® BRAND Certified Digital CASSETTES

TEN for $41
50+ $3.85

Won't drop a BIT!

$5.50

DISKETTES
8inch Soft (IBM)
8inch 32 sector
Mini Soft sec.
Mini 10 sector
Mini 16 sector

CALIFORNIA INDUSTRIAL is an Authorized Dealer of Scotch Brand Data Products

Shugart Associates

SA800-R Floppy Disk Drive

The most cost effective way to store data processing information, when random recall is a prime factor. The SA800 is fully compatible with the IBM 3740 format. Write protect circuitry, low maintenance & Shugart quality.

$449.50

CALIFORNIA DIGITAL 16 BiT 8086 S-100 CPU Board

Directly addresses one megabyte. 8 bit unidirectional & 16 bit bidirectional. 4K of static memory is supplied on board. $650.00

DigiCast A/V-100

R.F. MODULATOR

$29.95 Broadcast both audio and vidio on your existing color television. Recommended for the Apple II.

MEMORY

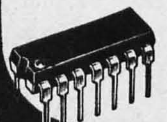
TRS-80 APPLE II $65
16k memory (8) 4116's

* As you may be aware, publishers require advertisers to submit their ad copy 60 to 90 days prior to "press" date. That much lead time in a volatile market place, such as memory circuits, makes it extremely difficult to project future cost and availability.
To obtain the best pricing on memory we have made volume commitments to our suppliers, which in turn affords us the opportunity to sell these circuits at the most competitive prices. Please contact us if you if you have a demand for volume state of the art memory products.

STATIC	1-31	32-99	100-5C	-999	1K+
21L02 450nS.	1.49	1.19	1.05	.95	.89
21L02 250nS.	1.69	1.49	1.45	*	*
2114 1Kx4 450	6.95	6.50	6.25	6.00	5.75
2114 1Kx4 300	8.95	8.50	8.00	*	*
4044 4Kx1 450	5.95	5.50	5.00	*	*
4044 4Kx1 250	9.95	9.50	9.00	*	*
4045 1Kx4 450	8.95	8.50	8.00	*	*
4045 1Kx4 250	9.95	9.50	9.00	*	*
5257 low pow.	7.95	7.50	7.05	6.75	6.45

SPECIAL CIRCUITS

Z80A 4 MHz.	24.95	AY5-1013A UART	4.95
8080A CPU	9.95	Floppy Disc Controllers	
8085	22.50	WD 1771 single D.	39.95
8086 Intel 16 bits	*	WD 1781 Double D	65.00
TMS 9900 16 bits	49.95	WD 1791 D/D 3740	*



EPROMS		1-15	16-63	64+
1702A	2K	4.95	4.50	4.00
2708	8K	9.95	9.50	9.00
2716 5v	16K	49.95	45.00	42.50
2532	32K	*	*	*

PORTABLE DATA ENTRY SYSTEM

$139.50

These used data terminals were originally designed for chain store inventory control and order entry systems. The operator enters the inventory control number, merchandise on hand and the unit price. After all pertinent data has been entered into the recorder, the main warehouse is telephoned, the handset is placed in the acoustic coupler and all the recorded information is transmitted back to the master computer. With a little imagination and one of these portable entry systems, you should be able to exchange programs and computer information with associates across the country. All units were removed from service in working condition. Original cost $2,500. Each system comes complete with:

- Portable Cassette Drive Unit
- Removable Entry Keyboard with LED Display
- Five Gould "D" NiCads
- Acoustical Coupler
- Battery Charger
- DB25 Cable
- Shoulder starp
- Full Documentation

MINIATURE SWITCHES



your choice

$.98	10	50	100	1k
	$.88	.81	.73	.66

SPDT Miniature Toggles

7101 C&K	ON-NONE-ON
7107 jbt	ON-OFF(mnt.ON)
7108 CK	ON-(moment.ON)
Rocker JBT ______	DPDT
Rotary	3P-4-Pos.
Rotary	3P-6-Pos.
Push B (N.O.)	$.39ea. 4/$1

DIP Switch

$1.49 ea.	10	25	100	1K
	$1.29	1.15	.97	.83

specify 4 7 or 8 pos.

SPST

Digital Cassette Drive

COMPUTER CONTROLED



BACK IN STOCK

$79.50

This precision I/O assembly features remote software controlled search capabilities. Two independent capstan drive motors allow the computer to control direction and speed of the transport.
The assembly consists of a Raymond cassette transport, chassis, motherboard and three edge cards: read/write, capstan drive & control card.
Current replacement valued at over $700.00. Schematics and complete documentation included. USED, but in excellent condition.

DISCOUNT Wire Wrap Center

IC SOCKETS

pin	wire wrap ea.	25	50	low profile ea.	25	50
8	—	—	—	17¢	16	15
14	37¢	36	35	18	17	16
16	38	37	36	19	18	17
24	99	93	85	36	35	34
40	169	155	139	63	60	58

KYNAR WIRE WRAP

50ft. $.98

500	1,000	11,000
$9.	$15.	$105.

Battery wire wrapping tool

$29.95 BW630

OK HOBBY WRAP-30 wire wrap & strip tool

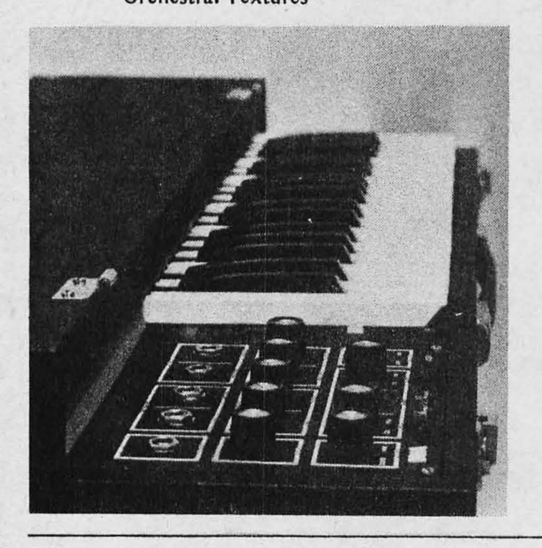
$5.45

(213) 679-9001

All merchandise sold by California Digital is premium grade. Sorry, no COD's. Orders are shipped the same day received. California residents add 6%. Foreign orders add 10%. Orders over $25, when accompanied by payment, are shipped at our expense. Otherwise, please add $2.

VISA master charge

Polytonic Keyboard System Generates Orchestral Textures



The 1550 Stringz-n-Thingz kit is a polytonic keyboard system from PAIA Electronics, 1020 W Wilshire Blvd, Oklahoma City OK 73116. This multipurpose instrument is capable of generating orchestral textures consisting of violin, cello, and piano voicings. A separate output also provides piano only, to allow separate processing, amplification, or mixing of the two types of voices. A full complement of operator controls allows switch selectable keyboard split, separate mixers for upper and lower keyboard, variable vibrato and chorusing rate and depth controls to allow reed organ and pipe organ voicings, and variable sustain controls for piano and strings.

A standard gate trigger jack allows the 1550 to be interfaced to any of the commercial synthesizers which feature systems interfacing jacks. This allows capabilities for brass synthesis, filtered strings, and other polytonic synthesizer effects.

Optional features include foot pedals for volume or sustain time control, foot switches for sustain control, and the 1551 stereo option to convert the mono string output to a true stereo output with two switch selectable modes of stereo operation. Other options include a processor interface to allow memorization of string or piano parts for later reproduction at any tempo and key desired. Also, the processor interface will allow the 1550 keyboard to simultaneously control a modular polyphonic synthesizer system.

The complete Stringz-n-Thingz kit including 84 page, step-by-step assembly and operation manual is available for $295.

Circle 604 on inquiry card.

Energy Monitor Saves Money

The Energy Monitor is an electronic device with a built-in microcomputer that budgets energy use on a daily basis and converts kW usage to visual dollars and cents. The amount of energy used is continually and automatically displayed in dollar amounts on a lighted digital display. Set a budget, and if usage exceeds the desired budgeted amount, a warning flashes. At the end of the billing period the Energy Monitor automatically resets to zero and starts a new month's computation. In addition to the savings on utility bills, consumers are entitled, under the Energy Tax Bill, to receive an investment tax credit for the purchase of this energy-saving tool.

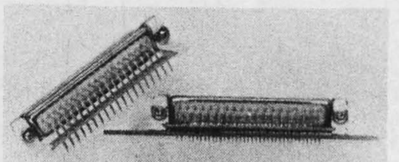
Installation is simple and inexpensive. A lighted panel displays eight functions: current dollar cost for energy used, projected amount of next bill, amount of last bill, billing date, chosen energy budget, cost per kW hour, date and time of day.



The unit retails for $295. For further information, contact Dupont Energy Management Corp, 3301 Conflans, Suite 102, Irving TX 75061.

Circle 605 on inquiry card.

Right Angle PC Mount Original D Connectors



An expanded range of right angle, printed circuit mount original D type subminiature connectors has been introduced by Souriau Inc, 7740 Lemona Av, Van Nuys CA 91405. The 831 series has been updated to include a fixed contact strap and a nonmetallic, fully insulated plastic mounting bracket and a new Underwriters Laboratory 94-VO rated thermoplastic insulator with a temperature range of −55°C to +105°C (−67°F to 221°F). The 831 series modifications are available in all layouts from 9, 15, 25, 37 to 50 pin and may be used with units from competitive manufacturers.

A comprehensive 12 page catalog detailing subminiature D, original D, D*M, and Norman/D connectors and accessories is available free upon request.

Circle 606 on inquiry card.

Computer Desk for Cromemco Computers



Cromemco is offering a new computer desk for the System Three and other Cromemco computers. The desk is styled and constructed to fit into any office surroundings or professional environment. The computer is mounted into a special shelf under the desk. This leaves the top free for a terminal, printer, or other unit, or as a work surface, while still providing the operator easy access to the computer for disk loading and unloading. The desktop is an attractive beige color designed to harmonize with the medium light wood veneer ends. The top surface is a tough, laminated plastic.

The Computer Desk (Model Z3-MDSK) is available for $695. For additional information, contact Cromemco Inc, 280 Bernardo Av, Mountain View CA 94043.

Circle 607 on inquiry card.

Standard Reference Alignment Cassette

Magnetic Information Systems has announced the introduction of an alignment reference metal cassette for use in the calibrating of digital and word processing equipment. This cassette is prerecorded at 1600 flux changes per inch (FCI) on an optical alignment recorder which employs precision magnetic heads. The magnetic tape is especially made for the digital reference tape application. These cassettes are priced at $12.50 and are available in several special configurations. For further information, contact Magnetic Information Systems Inc, 415 Howe Av, Shelton CT 06484.

Circle 608 on inquiry card.

Personal Computer Software Packages



GRT Corporation's G/2 personal computer software group has introduced 15 program packages containing 26 different programs for education, family entertainment, personal development and household data management. The G/2 line is compatible with the Radio Shack TRS-80, Apple II, Exidy Sorcerer, Processor Technology SOL and Southwest Technical Products 6800 computers.

The first available G/2 System Software includes two BASIC programming language packages by Microsoft. The G/2 Standard BASIC for the Southwest Tech 6800 computer is faster than Southwest Tech's BASIC, and offers 6800 owners a significantly broader selection of application programs. The G/2 Extended BASIC for the Processor Technology SOL is totally compatible with SOL's operating system, and provides features not available in Processor Technology BASICs.

Every G/2 product is produced on a tape cassette and packaged in a sturdy hard cover book style box along with an instruction manual. The application programs include source listings. The price for the G/2 personal computer program packages is $14.95; $34.95 for the Southwest Technical Products Standard BASIC; $49.95 for the SOL Extended BASIC. For further information, contact GRT Corp, 1286 Lawrence Station Rd, Sunnyvale CA 94086.

Circle 575 on inquiry card.

Time Series Analysis and Statistical Software Package for North Star

Potters Programs, 22444 Lakeland, St Clair Shores MI 48081 has announced a comprehensive time series analysis and statistical software package written in BASIC for the North Star floppy disk system. This package allows the user to load data into disk data files, edit the data, analyze it with a series of programs, and output the data in various convenient formats.

The analysis programs include a fast Fourier transform, auto correlation, cross correlation, distribution function, probability function, negative peak, positive peak, average, root mean squared, and various statistical and correlation calculations. The data can be recorded on disk from manual input or from analog to digital converters. It can be scanned and edited to remove obvious noise, and any part of the file can be analyzed by any of the methods. This package is available on North Star disk for $120.

Circle 576 on inquiry card.

Learn PET BASIC

PET BASIC Compleat consists of twenty lessons of PET BASIC, including all the major BASIC keywords, cursor control, screen editing, and use of the graphic characters. This two cassette tutorial is especially designed for beginning Commodore PET users. The 170 page manual which accompanies the cassettes is indexed for quick reference, three hole punched for easy review, and reproduces all data appearing on the screen (except PET's graphics). Quizzes and exercises add to the fun of learning how to use and program the Commodore PET. The package is priced at $39.95. For further information, contact ARESCO, POB 43, Audubon PA 19407.

Circle 577 on inquiry card.

Software for the PET and TRS-80

Speakeasy Software has announced the availability of consumer oriented software for the PET and TRS-80, in addition to the Apple versions. The titles fall into two categories: the Continuing Education Series, which includes financial analysis and transactional analysis; and the Home Entertainment Series with Warlords, Bulls and Bears, Sportstrivia, Microtrivia, and Kidstuff. For further information, contact Speakeasy Software Ltd, POB 1220, Kemptville, Ontario CANADA K0G 1J0.

Circle 578 on inquiry card.

Software and Hardware for Jolt and TIM Owners

Three new products have been announced by The 6502 Program Exchange, 2920 Moana, Reno NV 89509. The JAB (Jolt Adapter Board Kit) is a hardware device designed to interface the Jolt computer to the KIMSI S-100 interface. The $19 JAB Kit includes a manual and all parts except the Jolt connectors.

A program called ERAC (Editor and Resident Assembler Controller) was developed for users of the read only memory version of the Jolt Resident Assembler. ERAC allows source text and object code to be placed in programmable memory. Residing in 2 K bytes, ERAC is an extension of the RAP. A paper tape is available for $5 and the manual is $4.50.

LEDIP (Line Editor Program) is a compact line oriented text editor that readily lends itself to modification or expansion. LEDIP will output source text suitable for usage with the programmable read only memory version of the Jolt Resident Assembler. The paper tape is $2.75, the manual is priced at $3.25 and the cross assembly is $5.

Circle 579 on inquiry card.

Accounting Programs for Small Computers

The *Standard Software Library* is a series of books containing listings or programs written in BASIC with complete documentation. Each volume in the series is devoted to a single application.

The first three volumes deal with accounting programs for small computers. Volume 1 (General Ledger) enables a small business to set up a fully automated general ledger system with a complete chart of accounts. Included are programs for editing, sorting, merging and posting of transactions. A trial balance report is available in either summary or detail at the user's option. Income statement and balance sheet reports may be obtained at the close of each accounting period with both current and year-to-date totals and percentages. Volume 2 (Accounts Receivable) provides a fully automated system for dealing with customer accounts. Volume 3 (Payroll) enables a business to automate all of the normal payroll functions.

All of the programs are written in a level of BASIC which is common to almost all current microprocessors and minicomputers. The modular nature of the programs and the accompanying documentation make it easy to revise the program to meet special user requirements. The price of the Standard Software Library is $49.95. For further information, contact Creative Computer Consultants Inc, POB 2111, Norwalk CT 06852.

Circle 580 on inquiry card.

10-DAY FREE TRIAL

Send for our FREE Catalog



NCE COMPUMART'S PET WAREHOUSE

We have more PETs than anyone, call for quick delivery.



4K - Keyboard C	$ 595
8K - Keyboard C	$ 795
16K - Keyboard B	$ 995
16K - Keyboard N	$ 995
32K - Keyboard C	$1195
32K - Keyboard B	$1195
32K - Keyboard N	$1195

C — calculator keyboard (only version with tape deck)
B — large business keyboard without graphics symbols
N — large keyboard with graphics symbols

PET ACCESSORIES

Commodore Dual Floppy Disk Drive	$1295.00
New! PET Terminal Package	$69.00
Second Cassette — from Commodore	$95.00
Commodore PET Service Kit	$30.00
Beeper - Tells when tape is loaded	$24.95
Petunia - Play music from PET	$29.95
Video Buffer - Attach another CRT	$29.95
Combo - Petunia and Video Buffer	$49.95
New Serial Printer Interface for PET	$79.95
Integral Data Printer w/new interface	$878.95
PET - Compatible Selectric in Desk	$895.00
TTY KSR-33 Screen Printer for PET SALE!	$395.00
Originate/Answerback Modem for PET	$320.00
Bi-directional RS-232 Interface	$280.00
Betsi 4-slot S-100 Motherboard	$160.00
S-100 PET Interface was $289.00 SALE	$99.00

SAVE UP TO $100 ON READER/PUNCH

Our reconditioned paper tape Reader/Punches are reliable performers and now for a limited time they're all on sale! The serial Reader/Punch which was $950 is sale priced at $850. The Reader/Punch alone is marked down from $175 to $157, prices with interface follow:

Interface Model	Interface Added	Interface and Power	Interface, Power and Enclosure
Serial	$472	$616	$850
Parallel	$355	$598	$675

EXIDY SORCERER for $895

- Z-80 Processor
- Full-size ASCII keyboard
- User definable characters
- Up to 32K on-board RAM
- Interchangeable ROM PACs
- 30 x 64 display
- Resident 4K Monitor ROM
- Dual Cassette I/O
- Serial and Parallel I/O
- Complete Operators Manual
- Only $895

Top value in our catalog, simply add a monitor and tape decks to complete the system. Can be used as a terminal. Later ROM PACs will offer Z-80 Assembly Language, APL, PILOT, Word Processing and a DOS PAC for disk-based FORTRAN and COBOL. What a system for $895

Sorcerer 8K	$895
Sorcerer 16K	$1150
Sorcerer 32K	$1395
16K Expansion	$299

apple II

$200 FREE ACCESSORIES

Buy a 48K Apple II, mention this ad and take $200 in accessories free (if ordered together). This offer is good for $150 on 32K and $100 on 16K Apple II's. Now you can enjoy more of the best for less.

16K Apple II — $1195 (take $100 in free accessories)
32K Apple II — $1345 (take $150 in free accessories)
48K Apple II — $1495 (take $200 in free accessories)

Apple II Accessories

Centronics Printer Interface	$225
Disk and Controller	$595
Second Disk Drive	$495
Parallel Printer Card	$180
Communications Card	$225
Hi-Speed Serial Card	$195
Firmware Card	$200
Hobby/Proto Card	$24
Microverter RF Mod.	$35
Sanyo M2544 Recorder	$55



OF THE MONTH

NEW PET TERMINAL OPTION



- Get a PET & a terminal in one
- Data rates to 2400 baud
- Full Duplex operation
- Auto line wrap-around
- With or without auto line-feed

Buy any model PET & get a free NCE PET terminal option, normally $69. Now your PET can handle two different jobs. You can use the PET for a powerful BASIC computer or you can load our machine-code program to use the PET as a dumb terminal (not both at once). Please specify this option with order.

For those of you who already have a PET, you may order an NCE PET Terminal option separately for $69.

50% OFF CENTRONICS PRINTERS

Refurbished, excellent working condition. 90 day warranty, 10 day return privilege of course. While they last.

Model	Specifications	Centronics List	NCE Price
CENTRONICS 700	60 cps. 132 64 5 x 7 chars tractor feed	$1.520	$ 825
CENTRONICS 701	Same as 700, bidirectional	1.815	1,045
CENTRONICS 703	Bidirectional, 180 cps, 132 col. 64 7 x 7 chars tractor feed	2.805	1,695
CENTRONICS 779	60/100 cps, 80/132 col. 64 5 x 7 chars	1.250	625
CENTRONICS 780	60 cps, 80 col. 64 5 x 7 chars	1.905	1,095
CENTRONICS 781	Bidirectional, 60 cps, 80 col. 64 5 x 7 chars tractor feed	1.980	1,125
CENTRONICS 761-1 (KSR)	Keyboard send/receive, bidirectional 60 cps, 132 column, 64 7 x 7 chars tractor feed, 110/300/1200 baud	1.850	1,025
CENTRONICS 761-5 (RO)	Same as 761-1, receive only	1.750	945
		NCE Reg. Price	
CENTRONICS 101	165 cps, 132 col. 64 5 x 7 chars tractor feed	$1.500	1,125

Operator's Manual Included. For Technical Manual, add $15.00



Hazeltine 1400

IN STOCK NOW Only **$699.00**

Now a 5th Generation in Terminals: Immediate Delivery

Hazeltine 1410 with numeric keypad	$835
Hazeltine 1500 full of features SALE!	$999
Hazeltine 1510 with buffer logic	$1149
Hazeltine 1520 with printer interface	$1499

call or write for more information

SALE!

LOW-COST PRINTER FOR PET

This is a completely refurbished Teletype KSR-33 terminal ready to attach to your PET's interface to use as a printer

INTRODUCTORY PRICE $395

SALE!

TI PROGRAMMER

Hexadecimal, octal, decimal calculator/converter

Reg. $59.95 NOW $49.95

Datel/Selectric Table-top Terminals



- Local, remote or wait status selection
- Computer controllable terminal functions
- Remote interrogation of terminal status
- Interrupt and reverse break
- Interface RS232C (BCD code)



~~$395~~ $295 AS IS

CAT COUPLER

New 300 baud Originate/Answerback Acoustic Coupler. Looks good, works great and sale priced at **$189.**

NOW IN STOCK SYM BASIC $159

KIM-1

The original fully expandable single board computer

SALE Assembled **$169.95**

with power supply **$214.95**

EXPAND YOUR KIM SYSTEM:

KIMSI

KIM to S-100 Board	
Assembled	$169
Connector Set	15
KIM 4	
Expansion board	89.95
MICRO TECH. POWER SUPPLY Assembled	$34.95

KIM ENCLOSURE

Give your KIM a professional appearance. No alterations required, sturdy Kydex 100 Black **$23.50**

SUPER SALE List $279.95 **$189.95**

S-100 MPA

S-100 MPA gives your PET complete control of the S-100 bus (even DMA). Get an assembled unit at kit price.



SALE ~~$225~~

SANYO MONITOR

VM4209 9" quality monitor **$169**

NCE/CompuMart, Inc.

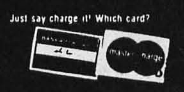
1250 North Main Street, Department BY69
P.O. Box 8610 Ann Arbor, Michigan 48107

10 DAY RETURN PRIVILEGE

(313) 994-3200

Just say charge it! Which card?

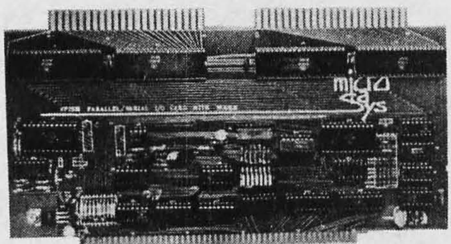




• Michigan residents add 4% sales tax • Add 4% shipping and handling for all domestic orders • Foreign orders (except Canada) add 10% additional handling (U.S. currency only) • P.O.'s accepted from D & B rated companies — shipment contingent upon receipt of signed purchase order • Phones open from 9:00 a.m. to 7:00 p.m. EST Monday-Friday, 10:00 a.m. to 5:00 p.m. Saturdays • Open accounts invited — call for credit application • Most items in stock for immediate shipment — call for delivery quotation • Sorry, no C.O.D.'s • All prices subject to change without notice • In the Ann Arbor area? Retail store open 11:00 a.m. to 7:00 p.m. Tuesday-Friday, 10:00 a.m. to 5:00 p.m. Saturdays (Closed Sunday and Monday)

SEND FOR FREE TERMINAL FLYER

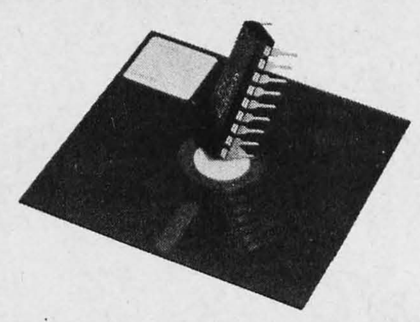
Parallel/Serial I/O Card with Modem



The 8P2SM Parallel/Serial I/O (input/output) card with modem is the latest product from MicroDaSys, POB 36051, Los Angeles CA 90036. It combines eight parallel ports (including full handshaking) with two serial input and output ports. It also enables the user to configure one set of serial ports for full RS-232 operation, and the other as a full duplex answer or originate modem. A complete documentation package is included with the board. The price is $149 in kit form and $199 completely assembled and tested.

Circle 598 on inquiry card.

Floppy Disk Read Amplifier System From Motorola



Motorola has introduced its MC3470 Floppy Disk Read Amplifier System. Combining both linear and digital functions on one integrated circuit, the MC3470 provides all signal processing from the read head through to the standard logic level digital output.

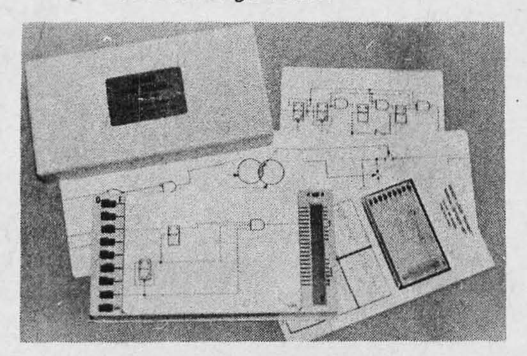
Contained in the circuit are the required gain stages, an active differentiator-comparator for peak detection and a time domain filter for wave shaping and elimination of false outputs. External connections for the required filter network, active differentiator and timing control components allow the system designer optimum flexibility in meeting overall system performance requirements.

This single monolithic device provides a standard TTL (transistor-transistor logic) digital output which is free from amplitude and waveform variations present at the read head, with a guaranteed maximum unadjusted peak shift of 5.0%.

The MC3470 Floppy Disk Read Amplifier is available in an 18 pin plastic dual-in-line package at the price of $5.95 for quantities of 100 and up. For more information, contact Motorola Semiconductors, POB 20912, Phoenix AZ 85036.

Circle 600 on inquiry card.

Self-Pace Logic Trainer



The Model 100 Broder Logic Trainer trains students without previous electronic background for digital electronic related assignments. It improves and grades the ability of the user. The Model 100 includes all gates, flip flops, positive and negative edge triggered devices, master-slave clocking, preset and clear functions. Switch circuit and Venn diagram problems as well as BCD and binary counting modes are included.

Physical logic state manipulation and the visual display make for fast and retained learning. In operation, the user manipulates component logic states using the eight logic switches. Solving a problem requires logic switch manipulation to force a logic 1 at the problem card output, which will turn on the designated bar indicator. A manual, 40 digital problems, and a 9 V battery cell are included. The Model 100 requires no wire or integrated circuit manipulation. It is priced at $69.95 and is available from L J Broder Enterprises Inc, 3192 Darvany Dr, Dallas TX 75220.

Circle 599 on inquiry card.

Apple II Software

The MUSE Co, POB 13365, Baltimore MD 21203 has announced a complete line of software for the Apple II computer. A full feature text editor ($17.95) allows management of free form text. Multiple space compression and tape I/O (input/output) are used for efficient file storage. U-Draw ($17.95) is a high resolution programmable graphics editor with tape I/O for storing finished drawings. Documentation includes instructions for linking figures to user programs. The Electric Crayon ($17.95) is a graphics editor similar to U-Draw but in low resolution color. The Music Box ($12.95) gives three octaves of sound with no additional hardware. Type in a song, and the Music Box will play it for you. Notation includes sharps, flats, note time, rests, dotted notes and tempo. It can be retuned easily for special sound effects. The Number Cruncher ($9.95) is a set of single precision math and ASCII to hexadecimal subroutines. Games which are priced at $12.95 each are also available.

Circle 601 on inquiry card.

DC Motor Speed Control in a Dual-In-Line Package

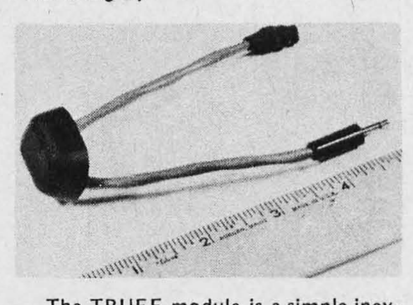
This monolithic integrated circuit DC motor speed control, housed in a 14 pin, low profile plastic dual-in-line package, is available from Cherry Semiconductor Corp, 3600 Sunset Av, Waukegan IL 60085. Designated the CS-175, the motor speed control is designed to provide maximum flexibility at a low cost. Requirements for adjustment and external components in multiple speed applications have been reduced by giving accurate, pin-programmable speed ratios for slow, medium, or fast motor velocities.

While many other applications are possible, the CS-175 is primarily intended for use with AC tachometer signals. The unit is capable of providing such stability that errors are dominated by terms created by the finite loop bandwidth made necessary to ensure stability with the dynamics of the specific motor and load.

The price for the CS-175 is $1.68; $.79 in 1,000 piece quantities; and $.65 in 10,000 quantities.

Circle 602 on inquiry card.

New Module Solves TRS-80 Cassette Drive Hang-Up



The TBUFF module is a simple inexpensive nonrepair shop solution for TRS-80 cassette drive hang-up. The module simply plugs in series with the remote cable between the TRS-80 and the recorder. TBUFF reduces the current passed through the reed relay in the TRS-80. At the same time, TBUFF delivers full power to the recorder, thus maintaining proper tape speed and volume levels. TBUFF sells for $7.95 (California residents add 6%). For further information, contact Web Associates, POB 60, Monrovia CA 91016.

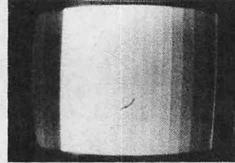
Circle 603 on inquiry card.

SOLID STATE SALES . . . *Announces a Breakthrough in Computer Technology*

A PICTURE MAY BE TAKEN BY OUR CAMERA, STORED IN A COMPUTER IN REAL TIME AND THEN DISPLAYED ON A CRT AT AN AFFORDABLE PRICE



VIDEO COMPUTER PROCESSING SYSTEM



GRAY LEVELS

THIS REMARKABLE VP-1 COMPUTER/ INTERFACE KIT HAS THE FOLLOWING:

FEATURES

- IT PRODUCES COMPOSITE VIDEO OUTPUT IN A 128 × 128 MATRIX FROM A DIRECT MONITOR CONNECTION USING 8K OF MEMORY
- THE SYSTEM USES A STANDARD S 100 BUSS
- WILL NOT TIE UP COMPUTER SOFTWARE WHEN NOT ADDRESSED
- IT DISPLAYS CONTINUOUSLY WHEN NOT ADDRESSED
- IT MAY PRODUCE PSEUDO COLOR AND/OR GRAPHICS (UP TO 16 GREY LEVELS, 4 BIT BINARY)

THE CAMERA WILL TAKE BETWEEN 15 AND 100 FRAMES/SECOND. THE CAMERA CONNECTS TO THE PROCESSOR WITH SEVEN LINES. THIS INCLUDES VIDEO AND TIMING SIGNALS

APPLICATIONS

- CONTINUOUS SURVEILLANCE
- INSPECTION OF MOVING PARTS WITH PROPER STROBING
- VISUAL GRAPHIC INPUT TO A COMPUTER
- CHARACTER OR PATTERN RECOGNITION
- PICTURES MAY BE TAKEN DIRECTLY FROM A TV WITHOUT ELECTRICAL CONNECTIONS
- THE INTERFACE KIT MAY BE USED SEPARATELY AS A 128 × 128 16 LEVEL GRAPHIC DISPLAY

OUR VP1 VIDEO SYSTEM CONSISTS OF THE FOLLOWING KITS:

- CCD 202C SOLID STATE VIDEO CAMERA KIT ASSEMBLED & TESTED $499.00
- VP-1 COMPUTER/VIDEO INTERFACE SYSTEM (3 BOARDS) ASSEMBLED & TESTED $999.00
- ASSEMBLED 8K MEMORY BOARD (OPTIONAL) $235.00

THIS VIDEO COMPUTER KIT CAN WORK WITH THE GE, REDICON, OR ANY OTHER 128 × 128 SENSOR CAMERA

REGULATED POWER SUPPLIES

POWER SYSTEMS # PS1111
115-230V 50/60 cy. in 5v DC at 35A out.
6"x 16¾"x 15¼" 26 lbs. shipping weight $55.00

POWER SYSTEMS # PS1106
115-230V 50/60 cy. in 12v DC at 15A out.
5"x 16¾"x 5" 19 lbs. shipping weight. $49.00
(OV PROTECT)

C/MOS (DIODE CLAMPED)

4001 –	.18	4019 –	.37	4049 –	.35	74C73 –	.65
4002 –	.18	4020 –	.90	4050 –	.35	74C74 –	.45
4006 –	.95	4021 –	.90	4051 –	.65	74C86 –	.40
4007 –	.18	4022 –	.90	4053 –	1.10	74C93 –	.75
4009 –	.37	4023 –	.18	4055 –	1.25	74C151	1.40
4010 –	.37	4024 –	.75	4066 –	.70	74C160	1.05
4011 –	.18	4025 –	.18	4069 –	.45	74C161	1.05
4012 –	.18	4027 –	.37	4071 –	18	74C174	1.05
4013 –	.29	4028 –	.80	4072 –	.21	74C175	1.05
4014 –	.75	4029 –	.95	4076 –	.97	74C192	1.20
4015 –	.75	4030 –	.33	4518 –	.95	74C193	1.20
4016 –	.29	4035 –	.97	74C00 –	.22	74C901	.48
4017 –	1.05	4042 –	.65	74C02 –	.22	74C902	.48
4018 –	.90	4044 –	.65	74C10 –	.27	74C914	1.70

1488-RS232 INTERFACE –	$1.40	2117-16K DYN RAM –	7.95
1489-RS232 INTERFACE –	1.25	UPD410-1-4K STAT. RAM –	9.95
2509-TRI STATE STAT. SR –	1.35	4116-2 –	9.95
2513-CHARAC. GEN. U.P. –	6.75	5280/2107B 4K DYN RAM –	3.40
2518-HEX 32 BIT SR –	2.20	TMS 4050L –	3.95
2522-STATIC SHIFT REG. –	1.95	5204 4K PROM –	4.95
2708-8K EPROM (450 ns) –	12.95	82S23 –	1.95
TMS 4050NL –	2.95	82S126 –	2.95
TMS-3409 NC 80 BITS DYN. S.R.	1.35	82S129 –	3.25
2716-(TI) –	35.95	AY5 1013 VART –	5.25
2102-1 (450 ns) –	.99	TR1 602B –	4.95
21L02-1 (450 ns) –	1.25	8703C TELEDYNE 8 BIT AD CONVERT –	9.95
MM5270 4K X1 DYN. –	3.45	PT 1482-B PSAT –	6.95
MK4008P –	1.95	8080A –	7.75
2101-1 254 x 4 STATIC –	2.45	8212 –	2.25
2111-1 256 x 4 STATIC –	3.45	8216 –	2.25
2112-1 256 x 4 STATIC –	2.75	8224 –	2.95
2114-4K STAT. RAM (450 ns) –	8.50		

8130 – 2.95 8T380 – 2.25 8834 – 2.25 8838 – 2.25
8131 – 2.95 8830 – 2.95 8837 – 2.25

CRYSTALS $3.45 ea.
2.000 MHz
4.000 MHz
5.000 MHz
6.000 MHz
8.000 MHz
10.000 MHz
20.000 MHz

RIBBON CABLE
FLAT (COLOR CODED)
#30 WIRE
26 cond. - .50/per foot
40 cond. - .75/per foot
50 cond. - .90/per foot

4" diameter .4V at 1 AMP $10.00
CTS 206-8 eight position dip switch $1.60
CTS 206-4 four position dip switch. $1.45
LIGHT ACTIVATED SCR's to 18. 200 V 1A. . $.70

SILICON SOLAR CELLS
2¼" diameter .4V at 500 ma $4.00

LED READOUTS

FND 359 C.C. .4" $ .60
FCS 8024 4 digit C.C. 8" display $5.95
FND 503 C.C. .5" $ .85
FND 510 C.A. .5" $ .85
DL-704 .3" C.C. $ .85
DL-704 C.A. .3" $ .75
DL 747 C.A. .6" $1.25
HP3400 .8"CC $1.95
HP3405 .8"CA $1.95

PRINTED CIRCUIT BOARD

4-1/2"×6-1/2" SINGLE SIDED EPOXY BOARD 1/16" thick, unetched
$.60 ea. . . . . 5/$2.60

7 WATT LD-65 LASER DIODE IR $8.95

2N 3820 P FET . . . . . . $ .45
2N 5457 N FET . . . . . . $ .45
2N2646 UJT $ .45
ER 900 TRIGGER DIODES 4 $1.00
2N 6028 PROG. UJT $ .65

MINIATURE MULTI-TURN TRIM POTS
100, 1K, 2K, 5K, 10K, 20K, 50K,
200K, 1Meg, 2Meg, $.75 each 3/$2.00

CHARGED COUPLE DEVICES

CCD 201C 100x100 Image Sensor $95.00
CCD 202C 100x100 Image Sensor $145.00

VERIPAX PC BOARD $4.00
This board is a 1/16" single sided paper epoxy board, 4½"x6½" DRILLED and ETCHED which will hold up to 21 single 14 pin IC's or 8,16 or LSI DIP IC's with busses for power supply connector.

FP 100 PHOTO TRANS . . . .$ .50
RED, YELLOW, GREEN LARGE LED's. 2" . . . . .6/$1.00
TIL-118 OPTO-ISOLATOR . . . .$ .75
MCT-6 OPTO ISOLATOR . . . .$ .80
1 WATT ZENERS: 3.3, 4.7, 5.1, 5.6, 9.1, 10, 12, 15, 18, or 22V . . . . . . 6/$1.00
MCM 6571A 7 x 9 character gen . . $ 10.75

UNIVERSAL 4Kx8 MEMORY BOARD KIT
$69.95

32-2102-1 fully buffered, 16 address lines, on board decoding for any 4 of 64 pages, standard 44 pin buss, may be used with F-8 & KIM

Silicon Power Rectifiers

PRV	1A	3A	12A	50A	125A	240A
100	.06	.14	.35	.90	3.70	5.00
200	.07	.20	.40	1.30	4.25	6.50
400	.09	.25	.65	1.50	6.50	9.50
600	.11	.30	.80	2.00	8.50	12.50
800	.15	.35	1.00	2.50	10.50	16.50
1000	.20	.45	1.25	3.00	12.50	20.00

SAD 1024 a REDICON 1024 stage analog "Bucket Brigade" shift register. $14.95

IN 4148 (IN914) 15/$1.00
.1 uf 25V ceramic caps 16/$1.00, $5.00/100

RS232 CONNECTORS
DB 25P male $2.95
DB 25S female $3.50
HOODS $1.50

REGULATORS

323K - 5V 3A . . $ 5.75
309K . . . .$ 1.10
723 . . . .$ .50
320T - 5, 12, or 15 V . . .$ 1.10
340K - 12, 15 or 24 V . . . .$ 1.10
340T - 5, 6, 8, 12 15, 18 or 24V $ 1.10
78 MG . . . . . .$ 1.35
79 MG . . . . . .$ 1.35

TRANSISTOR SPECIALS

2N6233-NPN SWITCHING POWER $ 1.95
MRF-8004 a CB RF Transistor NPN $ .75
2N3772 NPN Si TO-3 . . . . $ 1.00
2N1546 PNP GE TO-3 . . . .$ .75
2N4908 PNP Si TO-3 $ 1.00
2N5086 PNP Si TO-92 . . . . 4/$ 1.00
2N3137 NPN Si RF . . . . $ .55
2N3919 NPN Si TO-3 RF . . . . . . $ 1.50
2N1420 NPN Si TO 5 . . . . .3/$ 1.00
2N3767 NPN Si TO-66 . . . . $ .70
2N2222 NPN Si TO-18 . . . . 5/$ 1.00
2N3055 NPN Si TO-3 . . . . $ .50
2N3904 NPN Si TO-92 . . . . . . 6/$ 1.00
2N3906 PNP Si TO-92 . . . . . . 6/$ 1.00
2N5296 NPN Si TO-220 . . . . $ .50
2N6109 PNP Si TO-220. . . . . $ .55
2N3638 PNP Si TO-5 . . . . 5/$ 1.00
MPSA 13 NPN Si. . . . . 4/$ 1.00

TTL IC SERIES

		7446 –	.68	74153 –	.61
		7447 –	.58	74154 –	.94
7400 –	.15	7448 –	.68	74155 –	.58
7401 –	.15	7450 –	.18	74157 –	.55
7402 –	.15	7472 –	.25	74158 –	.65
7403 –	.15	7473 –	.28	74161 –	.55
7404 –	.18	7474 –	.28	74162 –	.80
7405 –	.18	7475 –	.45	74163 –	.55
7406 –	.22	7476 –	.30	74164 –	.85
7407 –	.24	7480 –	.31	74165 –	.95
7408 –	.18	7483 –	.65	74173 –	1.20
7409 –	.18	7485 –	.87	74174 –	.95
7410 –	.15	7486 –	.28	74175 –	.85
7411 –	.18	7489 –	1.25	74176 –	.75
7412 –	.18	7490 –	.42	74177 –	.75
7413 –	.36	7491 –	.58	74180 –	.65
7414 –	.60	7492 –	.43	74181 –	1.90
7416 –	.22	7493 –	.43	74190 –	1.00
7417 –	.25	7494 –	.67	74191 –	.85
7420 –	.18	7495 –	.65	74192 –	.79
7425 –	.30	7496 –	.65	74193 –	.79
7426 –	.22	74107 –	.28	74194 –	.80
7427 –	.28	74121 –	.30	74195 –	.50
7430 –	.18	74122 –	.40	74196 –	.86
7432 –	.22	74123 –	.45	74197 –	.80
7437 –	.22	74125 –	.40	74279 –	.55
7438 –	.22	74126 –	.40	74298 –	.90
7440 –	.18	74145 –	.65	74368 –	.62
7441 –	.70	74148 –	1.10	74393 –	1.25
7442 –	.45	74150 –	1.00	75322 –	1.50
7445 –	.65	74151 –	.61	74325 –	1.50

DATA CASSETTES 1/2 HR $ .95

14 pin headers 3/$1.00

MM 5387AA new clock chip which will directly drive LED's 12/24 hrs., 1 supply & alarm $5.95

NO. 30 WIRE WRAP WIRE SINGLE STRAND 100' $1.40

ALCO MINIATURE TOGGLE SWITCHES
MTA 106 SPDT . . . . $ .95
MTA 206 DPDT . . . . $ 1.70
MTA 206 P-DPDT CENTER OFF $ 1.85
MSD 206 P-DPDT CENTER OFF LEVER SWITCH . . . . $ 1.85

Full Wave Bridges

PRV	2A	6A	25A
100			1.30
200	.75	1.25	2.00
400	.95	1.50	3.00
600	1.20	1.75	4.00

DIP SOCKETS

8 PIN .17 24 PIN .35
14 PIN .20 28 PIN .40
16 PIN .22 40 PIN .60
18 PIN .25

SANKEN AUDIO POWER AMPS

Si 1010 G 10 WATTS. . . . . $ 7.80
Si 1020 G 20 WATTS. $15.70
Si 1050 G 50 WATTS. $28.50

TANTULUM CAPACITORS

.22UF 35V 5/$1.00	6.8UF 35V 4/$1.00
.47UF 35V 5/$1.00	10UF 10V $ .25
.68UF 35V 5/$1.00	22UF 25V $ .40
1UF 35V 5/$1.00	15UF 35V 3/$1.00
2.2UF 20V 5/$1.00	30UF 6V 5/$1.00
3.3UF 20V 4/$1.00	33UF 20V $ .40
4.7UF 15V 5/$1.00	47UF 20V $.35
	68UF 15V $ .50

74LS SERIES

74LS00 –	.31	74LS138 –	.94
74LS02 –	.31	74LS139 –	.94
74LS03 –	.31	74LS151 –	.88
74LS04 –	.34	74LS153 –	.88
74LS05 –	.34	74LS155 –	.92
74LS08 –	.34	74LS156 –	.92
74LS09 –	.34	74LS157 –	.92
74LS10 –	.31	74LS160 –	1.21
74LS11 –	.31	74LS161 –	1.21
74LS13 –	.59	74LS162 –	1.21
74LS14 –	1.08	74LS163 –	1.21
74LS15 –	.31	74LS164 –	1.34
74LS20 –	.31	74LS168 –	1.42
74LS21 –	.31	74LS169 –	1.42
74LS22 –	.31	74LS170 –	1.89
74LS26 –	.39	74LS173 –	1.62
74LS27 –	.39	74LS174 –	1.35
74LS28 –	.39	74LS175 –	1.35
74LS30 –	.31	74LS181 –	2.97
74LS32 –	.39	74LS190 –	1.48
74LS37 –	.37	74LS191 –	1.48
74LS38 –	.37	74LS192 –	1.19
74LS40 –	.31	74LS193 –	1.48
74LS42 –	.88	74LS195 –	1.08
74LS51 –	.34	74LS196 –	1.08
74LS54 –	.34	74LS197 –	1.08
74LS73 –	.47	74LS241 –	3.37
74LS75 –	.67	74LS257 –	1.19
74LS76 –	.47	74LS258 –	1.19
74LS86 –	.47	74LS259 –	1.82
74LS90 –	.67	74LS265 –	.40
74LS92 –	.67	74LS266 –	.53
74LS93 –	.67	74LS279 –	.94
74LS109 –	.47	74LS290 –	.81
74LS112 –	.47	74LS293 –	.78
74LS113 –	.47	74LS365 –	.81
74LS114 –	.47	74LS366 –	.81
74LS124 –	1.25	74LS368 –	.81
74LS125 –	.61	74LS375 –	.94
74LS126 –	.61	74LS386 –	.47
74LS132 –	.94	74LS390 –	.196
		74LS670 –	2.70

LINEAR CIRCUITS

LM 101 –	.75
LM 301/748	.25
LM307 –	.30
LM 308 –	.75
LM 311 –	.75
LM 318 –	1.20
LM 324 –	.70
LM 339 –	1.10
LM 358 –	.70
LM 370 –	1.15
LM 377 –	1.60
LM 380 –	.95
LM 382 –	1.25
LM386 –	.80
LM 387 –	1.25
LM 537 –	2.50
LM 553 –	2.50
LM 555 –	.49
LM 556 –	.85
NE540L –	2.25
560 –	2.00
565 –	.95
566 –	1.10
567 –	1.10
703 –	.90
733H –	.75
709 –	.25
710 –	.35
711CH –	.40
741C or V –	.25
747 –	.50
LM 1310 –	2.50
1456 –	.95
1458 –	.50
3900 –	.40
8038CC –	3.90
791 –	1.95
LF356H –	1.20

TRIACS / SCR'S

PRV	TRIACS 1A	TRIACS 10A	TRIACS 25A	SCR'S 1.5A	SCR'S 6A	SCR'S 35A
100	.40	.70	1.30	.40	.50	1.20
200	.70	1.10	1.75	.60	.70	1.60
400	1.10	1.60	2.60	1.00	1.20	2.20
600	1.70	2.30	3.60		1.50	3.00

Terms: FOB Cambridge, Mass. Send Check or Money Order. Include Postage, Minimum Order $5.00, COD'S $20.00

Send 25¢ for our catalog featuring Transistors and Rectifiers
145 Hampshire St., Cambridge, Mass.



SOLID STATE SALES
P.O. BOX 74B
SOMERVILLE, MASS. 02143 TEL. (617) 547-7053

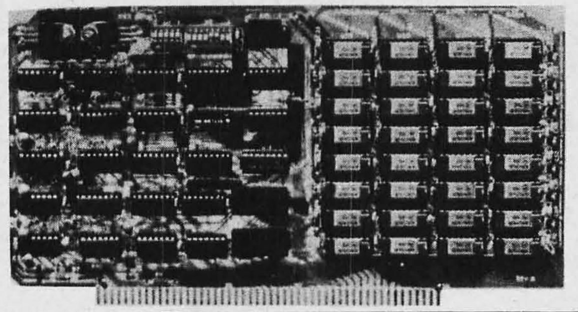
WE SHIP OVER 95% OF OUR ORDERS THE DAY WE RECEIVE THEM

Circle 340 on inquiry card.

Visit our new retail location!

SD EXPANDORAM

The Ultimate S-100 Memory



The EXPANDORAM is available in versions from 16K up to 64K, so for a minimum investment you can have a memory system that will grow with your needs. This is a dynamic memory with the invisable on-board refresh, and IT WORKS!

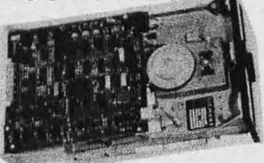
- Bank Selectable
- Phantom
- Power 8VDC, ±16VDC, 5 Watts
- Lowest Cost Per Bit
- Uses Popular 4116 RAMS
- PC Board is doubled solder masked and has silk-screen parts layout.

- Extensive documentation clearly written
- Complete Kit includes all Sockets for 64K
- Memory access time: 375ns, Cycle time: 500ns.
- No wait states required.
- 16K boundries and Protection via Dip Switches
- Designed to work with Z-80, 8080, 8085 CPU's.

EXPANDO 64 KIT (4116)

16K	$245.00
32K	$310.00
48K	$375.00
64K	$440.00

DISC DRIVES



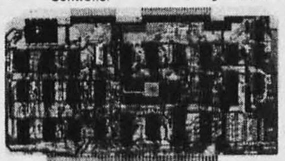
Sugart SA400 5¼"
with attractive metal case
$295.00

Sugart 801
with attractive metal case
$495.00

Siemens FDD 200-8 8" double-sided double density
$599.00

DISC CONTROLLER
SD "VERSAFLOPPY" Kit

The Versatile Floppy Disk Controller **Only $159.00**



FEATURES: IBM 3740 Soft Sectored Compatible. S-100 BUS Compatible for Z-80 or 8080. Controls up to 4 Drives (single or double sided). Directly controls the following drives:
1. Shugart SA400/450 Mini Floppy
2. Shugart SA800/850 Standard Floppy.
3. PERSCI 70 and 277.
4. MFE 700/750.
5. CDC 9404/9406.
6. GSI/Siemans FDD120-8.

34 Pin Connector for Mini Floppy. 50 Pin Connector for Standard Floppy. Operates with modified CP/M operating system and C-Basic Comptler. The new "Versafloppy" from S.D. Computer Products provides complete control for many of the available Floppy Disk Drives, Both Mini and Full Size. FD1771B-1 Single Density Controller Chip. Listings for Control Software are included in price.

SAVE $100.00

DM2700S DISK & CABINET with POWER SUPPLY

DM2700S includes Siemans FD120-8" Disk Drive with the following features:

- Single or Double Density
- Hard or Soft Sector
- Door Interlock
- Write Protect
- Hard Sector Detection
- 500 KB/S Transfer
- 800 KB unformated
- Bit density 6536 BP1
- Sugart 800 Series Compatable

Cabinet includes:
- 110V to 125V 60 Hz power supply
- Data Cable
- Fan
- Accepts per SCI, Shugart, Siemens 8" Drives

SHOP & COMPARE!

DM2700S Disk Drive & Cabinet
REG. $750 SALE PRICED **$650.00**

DM2700 Cabinet, less Drive ~~$249.00~~ **$225.00**

CONTINENTAL SPECIALTIES CORPORATION



Logic Probes and Digital Pulsers

SALE



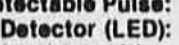
LOGIC PROBES

CSC logic probes are the ultimate tool for breadboard design and testing. These hand-held units provide an instant overview of circuit conditions. Simple to use; just clip power leads to circuit's power supply, set logic family switch to TTL/DTL or CMOS/HTL. Touch probe to test node. Trace logic levels and pulses through digital circuits. Even stretch and latch for easy pulse detection. Instant recognition of high, low or invalid levels, open circuits and nodes. Simple, dual-level detector LEDs tell it quickly, correctly. HI (Logic "1"); LO (Logic "0"). Also incorporates blinking pulse detector. e.g., HI and LO LEDs blink on or off, tracking "1" or "0" states at square wave frequencies up to 1.5 MHz. Pulse LED blinks on for ⅓ second during pulse transition. Choice of three models to meet individual requirements; budget, project and speed of logic circuits.

MODEL LP-1

Hand-held logic probe provides instant reading of logic levels for TTL, DTL, HTL or CMOS. **Input Impedance:** 100,000 ohms. **Minimum Detectable Pulse:** 50 ns. **Maximum Input Signal (Frequency):** 10 MHz. **Pulse Detector (LED):** High speed train or single event. **Pulse Memory:** Pulse or level transition detected and stored.

CSC Model LP-1 Logic Probe—Net Each .... ~~$44.95~~ **$42.70**



MODEL LP-2

Economy version of Model LP-1. Safer than a voltmeter. More accurate than a scope. **Input Impedance:** 300,000 ohms. **Minimum Detectable Pulse:** 300 ns. **Maximum Input Signal (Frequency):** 1.5 MHz. **Pulse Detector (LED):** High speed train or single event. **Pulse Memory:** None.

CSC Model LP-2 Logic Probe—Net Each .... ~~$24.95~~ **$23.70**

MODEL LP-3

High speed logic probe. Captures pulses as short as 10 ns. **Input Impedance:** 500,000 ohms. **Minimum Detectable Pulse:** 10 ns. **Maximum Input Signal (Frequency):** 50 MHz. **Pulse Detector (LED):** High speed train or single event. **Pulse Memory:** Pulse or level transition detected and stored.

CSC Model LP-3 Logic Probe—Net Each .... ~~$69.95~~ **$66.45**

SALE

DIGITAL PULSER

The ultimate in speed and ease of operation. Simply connect clip leads to positive and negative power, then touch DP-1's probe to a circuit node; automatic polarity sensor detects circuit's high or low condition. Depress the pushbutton and trigger an opposite polarity pulse into the circuit. Fast troubleshooting includes injecting signals at key points in TTL, DTL, CMOS or other popular circuits. Test with single pulse or 100 pulses per second via built-in dual control push-button; button selects single shot or continuous modes. LED indicator monitors operating modes by flashing once for single pulse or continuously for a pulse train. Completely automatic, pencil-size lab/field pulse generator for any family of digital circuits. **Output:** Tri-state. **Polarity:** Pulse-sensing auto-polarity. **Sync and Source:** 100 mA. **Pulse Train:** 100 pps. **LED Indicator:** Flashes for single pulse; stays lit for pulse train.

CSC Model DP-1 Digital Pulser—Net Each .... ~~$74.95~~ **$71.20**

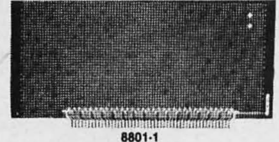
S100 PLUG BOARDS

Vector

4607

DEC LSI-11, PDP8, PDP11, Heath H-11, P Pattern Epoxy Glass, Plug Board 8.43"x5.187". Dual 36 pin DEC/HEATH Connectors.

1-4	5-9	10-24
19.95	17.96	15.96



8801-1

Plain no etched circuitry except contacts. Produces maximum flexibility.

1-4	5-9	10-24
14.95	13.46	11.96

8802-1

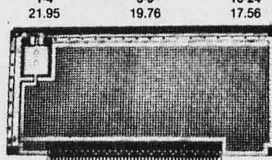
Pad per 2 holes. Two-holE pads allow tack soldering of socket, plus second hole for component leads.

1-4	5-9	10-24
21.95	19.76	17.56

8804

"ANY DIP" has full power and ground planes back to back. Board accommodates 3, 4, 6, 9" Dips.

1-4	5-9	10-24
21.95	19.76	17.56



8801

Individual tinned square pads surround most holes. Ideal for mounting components by "tack soldering" Top of board pod free for mounting I/O connectors.

1-4	5-9	10-24
19.95	17.95	15.96

Vector

4608

Is form and size compatible with INTEL SBL80 Series and NATIONAL BLC 80 Series microcomputer boards. Power and Ground buses on both sides.

1-4	5-9	10-24
45.00	40.50	36.00

4608-1

Same as 4608, except plain less power buses.

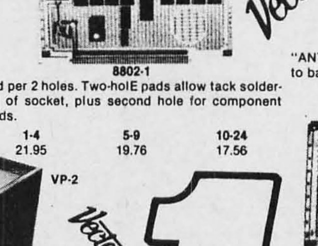
104	5-9	10-24
34.00	30.60	27.20

VECTOR-PAK ASSEMBLED MICROCOMPUTER CASES

Adjustable packaging system for S-100 bus microcomputers, compatible with Altair 8800 and IMSAI 8080 size cards.
- Smart looking, deluxe cases unmarred by unsightly screws or fasteners.
- Finished in dark blue textured vinyl.
- Instantly accessible interiors with slip out covers.
- Removable recessed rear and front panels.
- Fully adjustable interior mounting systems for any card or card spacing within size limitations. No cutting or drilling necessary.
- Perforated bottom cover for cooler operation.

DESCRIPTION

Assembled case with perforated bottom cover. Installed mounting struts for card guides and receptacles or mother board. Cards top loaded, spanning front to back. Card guide (12 pair) and chassis plate supplied uninstalled.



VP-2

Vector

VP-1
$163.00
Shipping Weight 25 lbs.

VP-2
$159.00
Same as VP1 except cards oriented side to side.

PRIORITY ONE ELECTRONICS ©

16723B Roscoe Blvd. Sepulveda, CA 91343

Terms: Visa, MC, BAC, Check, Money Order, C.O.D. U.S. Funds Only. CA residents add 6% sales tax. Minimum order $10.00. Prepaid U.S. orders less than $75.00 include 5% shipping and handling, minimum $2.50. Excess refunded. Just in case . . . please include your phone no.
Prices subject to change without notice.
We will do our best to maintain prices thru June 1979.

phone orders welcome (213) 894-8171, (800) 423-5633

OEM and Institutional inquiries invited.

ORDER TODAY 1-800-423-5633

Visit our new retail location!

Circle 312 on inquiry card.

HICKOK LX303 $74.95

HICKOK LX303

74^{95} *

.5%, 3½ digit 19 Range DVM. ½" LCD displays runs 200 hrs on 1 battery. 10 Meg Ohm Input. 1 yr. guarantee, made in U.S.A., test leads included.

Available Accessories

- **RC-3** 115V AC Adapter **$7.50**
- **CC-3** Deluxe Padded Vinyl Carrying Case **$7.50**
- **VP-10** X10 DCV Probe Adapter/ Protector 10Kv **$14.95**
- **VP-40** 40Kv DC Probe **$35.00**
- **CS-1** 10 Amp Current Shunt **$14.95**

*FREE

Just for Asking.
FREE BATTERY with your meter.





$318.
MS-15 MINISCOPE

With Rechargeable Batteries & Charger Unit

- 15 megahertz bandwidth.
- External and internal trigger.
- Time base — .1 microsec. to 0.5 Sec/div - 21 settings . ± 3%.
- Battery or line operation.
- Automatic & line sync modes.
- Power consumption < 15 watts.
- Vortical Gain — .01 to 50 V/div · 12 settings ± 3%.
- Viewing area 1.1" x 1.35".
- Case size 2.7"H x 6.4"W x 7.5"D, 3 pounds.
- Parts & Labor guaranteed 1 year
- 10 to 1, 10 meg probe .... $ 27.
- Leather carrying case .... $ 45.

PROBE 1¢

PROBE 1¢ with the purchase of SCOPE and the MENTION of this MAGAZINE

- **MS-215 Dual Trace Version of MS-15 $435.**

UNGARmatic®
Controlled Soldering Station

THREAD-TOGETHER MODULAR DESIGN FOR QUICK, ON-LINE HEATER OR TIP CHANGE

- Available in 3 preset temperatures; 600°F., 700°F. or 800°F. for any application.
- Closed loop, non-magnetic control.
- Low voltage system; 3 wire grounded.
- Biomechanical designed handle with cool grip for operator comfort. Cord is super flexible 3 wire grounded, heat resistant.
- Large capacity snap-on tray and sponge; removable for optional placement.
- Long life interchangeable tips, iron clad, chrome plated, pre-tinned.
- Designed for use on sensitive components.

COMPLETE STATION AND REPLACEABLE HEATER SELECTION GUIDE

Temperature	Complete Station	Controlled Heater No.
600°F.	50T6	#76
700°F.	50T7	#77
800°F.	50T8	#78

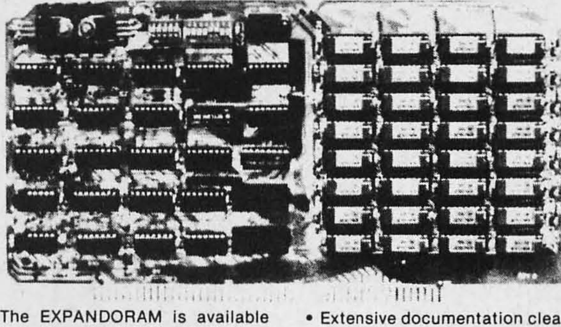
Each of the above stations include: #70B Power Supply with on/off switch, indicator light, and 3 wire power cord; #71 Handle with 3 wire heat resistant secondary cord; #89 Tray and Sponge; #72 Iron Holder; Controlled Heater with #87 Screwdriver tip 1/16".

#95 TIP ADAPTER Tip Adapter #95 for special micro applications. This Adapter is designed for the use of 1/8" thread-in Princess Nibs.

Controlled Heaters: $14.95 Each. Tips: $2.25 Each. Adapter: $1.30 Each

List $59.50 SALE PRICE $49.95

SALE SD EXPANDORAM
The Ultimate S-100 Memory



The EXPANDORAM is available in versions from 16K up to 64K, so for a minimum investment you can have a memory system that will grow with your needs. This is a dynamic memory with the invisable on-board refresh, and IT WORKS!

- Bank Selectable
- Phantom
- Power 8VDC, ± 16VDC, 5 Watts
- Lowest Cost Per Bit
- Uses Popular 4116 RAMS
- PC Board is doubled solder masked and has silk-screen parts layout.
- Extensive documentation clearly written
- Complete Kit includes all Sockets for 64K
- Memory access time: 375ns, Cycle time: 500ns.
- No wait states required.
- 16K boundries and Protection via Dip Switches
- Designed to work with Z-80, 8080, 8085 CPU's.

EXPANDO 64 KIT (4116)

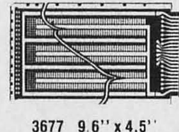
16K	$245.00
32K	$310.00
48K	$375.00
64K	$440.00

8803 Vector
MOTHER BOARD FOR S100 BUS MICRO-COMPUTERS

- Kit includes 12 tantalum capacitors for +5, +12, —12 buses and insulated mounting spacers.
- Wiring side shown. Component side bare epoxy glass with white markings for component locations.
- G10 epoxy glass board with 2 ounce copper, solder plated and .038 diameter holes for leads.
- Solder mask with solder windows on etched circuits to avoid accidental short circuits.
- Mounts 11 receptacles with 100 contacts (2 rows) on .125 centers with .250 row spacing, Vector part number R681-2, or mounts 10 receptacles plus interconnections to smaller mother board for expansion.
- Includes etched circuits and instructions for option of active, pull-up, or floating terminations.
- Large buses: +5V and GND (10 AMPS), ± 12V or 16V (7 AMPS). Current ratings are per MIL-STD-275 with 10°C rise.
- Fits in Vector-pak enclosures.
- Fits in IMSAI 8080 microcomputer as expander board.

Price: $29.50

Vector Plugboards



8800V

Universal Microcomputer/processor plugboard, use with S-100 bus. Complete with heat sink & hardware. 5.3" x 10" x 1/16".

1-4	5-9	10-24
$19.95	$17.95	$15.96

8801-1

Same as 8800V except plain, less power buses & heat sink.

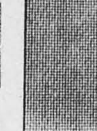
1-4	5-9	10-24
$14.95	$13.46	$11.96

3682 9.6" x 4.5" **$10.97**
3682-2 6.5" x 4.5" **$9.81**

Hi-Density Dual-In-Line Plugboard for Wire Wrap with Power & Grd. Bus Epoxy Glass 1/16" 44 pin con. spaced .156

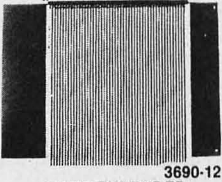
3677 9.6" x 4.5" **$10.90**
3677-2 6.5" x 4.5" **$9.74**

Gen. Purpose D.I.P. Boards with Bus Pattern for Solder or Wire Wrap. Epoxy Glass 1/16" 44 pin con. spaced .156



3662 6.5" x 4.5" **$7.65**
3662-2 9.6" x 4.5" **$11.45**

P pattern plugboards for IC's Epoxy Glass 1/16" 44 pin con. spaced .156



3690-12
CARD EXTENDER

Card Extender has 100 contacts 50 per side on .125 centers-Attached connector-is compatible with S-100 Bus Systems. **$25.83**
3690 6.5" 22/44 pin .156 ctrs. Extenders $13.17

1/16 Vector BOARD
.042 dia holes on 0.1 spacing for IC's



Phenolic PART NO.	SIZE	PRICE 1-9	PRICE 10-19
64P44XXXP	4.5x6.5"	$1.56	$1.40
169P44XXXP	4.5x17"	$3.69	$3.32
Epoxy Glass			
64P44	4.5x6.5"	$1.79	$1.61
84P44	4.5x8.5"	$2.21	$1.99
169P44	4.5x17"	$4.52	$4.07
169P84	8.5x17"	$8.03	$7.23

TRS-80/APPLE
MEMORY EXPANSION KITS
4116's RAMS
(16Kx1 200ns)

8 for $69.00

2708
8K 450 ns

EPROM
FACTORY PRIME

$ 9.00 ea.
25 + *Call For Price*

WOW 14 & 16 PIN GOLD 3 LEVEL WIRE WRAP SOCKETS



14 - G3 100 for **$30.00**
16-G3 100 for **$30.00**
50 of each for **$32.00**

Sockets are End & Side stackable, closed entry

PANAVISE



2114N-3L
4096 - BIT (1024x4) 300 ns Low Power Static Ram

8 for 50^{00}

ORDER TOLL FREE
1-800-423-5633
except CA., AK., HI., Call (213) 894-8171



2102LPC
450ns Low Power RAMS
$1.00 Ea. in lots of 25

2102LHPC
250ns Low Power RAMS
$1.25 Ea. in lots of 25

TMS 4044/ MM 5257-3L
4096x1 300 ns Low Power Static Ram

8 for 50^{00}



WRAP POST for .042 dia. holes (all boards on this page)
T44/C pkg. 100 . . **$ 2.34**
T44/M pkg. 1000 ..... **$14.35**
A-13 hand installing tool .... **$ 2.94**



PRIORITY ONE ELECTRONICS ©
16723B Roscoe Blvd. Sepulveda, CA 91343

Terms: Visa, MC, BAC, Check, Money Order, C.O.D. U.S. Funds Only. CA residents add 6% sales tax. Minimum order $10.00. Prepaid U.S. orders less than $75.00 include 5% shipping and handling, minimum $2.50. Excess refunded. Just in case . . . please include your phone no. Prices subject to change without notice. We will do our best to maintain prices thru June 1979.

phone orders welcome (213) 894-8171, (800) 423-5633

OEM and Institutional inquiries invited.

2716
5 volt only 16K EPROM

50^{00}

HICKOK LX303 $74.95 HICKOK LX303 $74.95

HICKOK LX303 $74.95 HICKOK LX303 $74.95 HICKOK LX303 $74.95 HICKOK LX303 $74.95 HICKOK LX303 $74.95 HICKOK LX303 $74.95

We have 5V 2716's in Stock. We have the Best Prices on 2102's, 2114's, 4116's.

S-100 BUS EDGE CONNECTORS

S100-WWG 50/100 Cont. .125 ctrs. 3 LEVEL WIRE WRAP .025" sq. posts on .250 spaced rows. GOLD PLATED.

1-4	5-9	10-24
$4.00	$3.75	$3.50

S100ALT 50/100 Cont. .125 ctrs. DIP SOLDER TAIL on .140 spaced rows for ALTAIR motherboards. GOLD plated.

1-4	5-9	10-24
$4.00	$3.75	$3.50

S100-STG 50/100 Cont. .125 ctrs. DIP SOLDER TAIL on .250 spaced rows for VECTOR and MASI motherboards GOLD plated.

1-4	5-9	10-24
$3.50	$3.25	$3.00

S100SE 50/100 Cont. .125 ctrs. PIERCED SOLDER EYELET tails. GOLD

1-4	5-9	10-24
$5.00	$4.50	$4.25

OTHER POPULAR EDGE CONNECTORS

All Edge Card Connectors are GOLD PLATED (not Gold Flash) Bodies are non brittle, Solvent res., G.E. Valox. Contacts are Bifurcated; Phos/Bronze: GOLD over Nickel.

ABBREVIATIONS: SE = Solder Eyelet WW = 3 Level Wire Wrap ST = Solder Tail

.100" Contact Center Connectors

PART NO.	TYPICAL APPLICATION	PRICE 1-4	5-9	10-24
D1326-1SE	Imsai M10, SIO	2.60	2.40	2.20
D2244-1WW	Vector Plugboards	4.00	3.80	3.60
D2250-1SE	Imsai P10, Intel Multibuss	3.70	3.50	3.40
D2250-1ST	Imsai P10, Intel Multibuss	3.50	3.30	3.10
D2040-1SE	TRS-80	3.20	3.05	2.90
D2040-1ST	TRS-80	3.00	2.85	2.70
D2040-1WW	TRS-80	3.30	3.15	3.00
D3060-1WW	Intel Multibuss	4.10	3.90	3.70
D3672-1SE	Vector Plugboards	5.00	4.75	4.50
D3672-1ST	Vector Plugboards	4.95	4.70	4.45
D3672-1WW	Vector Plugboards	4.90	4.65	4.40
D4080-1SE	PET	5.95	5.70	5.45
D4080-1ST	PET	5.00	4.75	4.50
D4080-1WW	PET	5.20	4.95	4.70
D4386-1SE	Cos. ELF	5.60	5.35	5.05
D4386-1ST	Cos. ELF	5.40	5.15	4.90
D4386-1WW	Cos. ELF	5.50	5.25	5.00
D50100-1WW		5.95	5.75	5.55

.125" Contact Center Connectors

PART NO.	TYPICAL APPLICATION	PRICE 1-4	5-9	10-24
D3672-2WW		5.25	5.00	4.75
D4080-2WW	Vector 4350	5.95	5.65	5.35
S100-STG	S-100, Imsai, Vector, Cromenco, Motherboards	3.50	3.25	3.00
S100-WWG	S-100 Wire Wrap	4.00	3.75	3.40
S100-ALT	Altair	4.50	4.25	4.00

.156 Contact Centers Connectors



PART NO.	TYPICAL APPLICATION	PRICE 1-4	5-9	10-24
S6X-5SE	Pet, NSC CLK Modules	1.40	1.30	1.20
D612-5SE	Pet, NSC CLK Modules	1.60	1.50	1.40
D1224-5SE	Pet	2.40	2.30	2.20
D2224-5ST	Pet	2.30	2.20	2.10
D1530-5SE	Vector Plugboards, GRI Keybrds	2.50	2.35	2.10
D1530-5ST	Vector Plugboards, GRI Keybrds	2.40	2.25	2.05
D1530-5WW	Vector Plugboards, GRI Keybrds	2.60	2.40	2.15
D1836-5SE		3.00	2.80	2.60
D2244-5SE	Vector, Kim, etc.	3.00	2.60	2.20
D2244-5ST	Vector, Kim, etc.	3.00	2.80	2.50
D2244-5WW	Vector, Kim, etc.	3.95	3.70	3.40
D3672-5SE	Vector Plugboards	5.50	5.30	5.00
D3672-5ST	Vector Plugboards	5.45	5.25	5.00
D3672-5WW	Vector Plugboards	5.60	5.40	5.10
DE4386-5SE	Mot 6800, Intel Multibuss, NSC pacer	6.00	5.75	5.25
D4386-5ST	Mot 6800, Intel Multibuss, NSC pacer	5.90	5.65	5.15
D4386-5WW	Mot 6800, Intel Multibuss, NSC pacer	6.50	6.30	5.90
CG-1	Imsai Style Card Guides	5/1.00 or 100/10.00		

RS232 & "D" TYPE CONNECTORS

P = Plug-Male S = Socket-Female C = Cover-Hood

PART NO.	DESCRIPTION	PRICE 1-4	5-9	10-24
DE-9P	9 Pin Male	1.50	1.30	1.10
DE-9S	9 Pin Female	1.95	1.75	1.45
DE-9C	9 Pin Cover	1.50	1.30	1.10
DA15P	15 Pin Male	2.00	1.80	1.55
DA15S	15 Pin Female	2.90	2.70	2.45
DA15C	15 Pin Cover	1.80	1.60	1.30
DB-25P	**25 Pin Male**	2.50	2.20	2.05
DB-25S	**25 Pin Female**	3.50	3.10	2.95
DB51212-1	1 pc. Grey Hood	1.65	1.35	1.20
DB1226-1A	2 pc. Black Hood	1.80	1.50	1.35
DB110963-3	2 pc. Grey Hood	1.70	1.40	1.25
DC37P	37 Pin Male	3.95	3.75	3.50
DC37S	37 Pin Female	5.50	5.25	4.90
DC37C	37 Pin Cover	2.00	1.80	1.60
DD50P	50 Pin Male	5.00	4.75	4.60
DD50S	50 Pin Female	6.50	6.00	5.75
DD50C	50 Pin Cover	2.50	2.30	2.20
D20418-S	Hardware Set (2 pair)	1.00	.80	.70

MALE

FEMALE

Connector for CENTRONICS 700 SERIES:
Amphenol 57-30360 for back of Centronics 700 Series printers 1-4—$9.00 5-up—$7.50

DIP PLUGS

Part #	No. of Pins	PRICE 1-24	25-99	100-499
P08P02	8	.41	.36	.29
P14P02	14	.48	.42	.34
P16P02	16	.55	.47	.38
P18P02	18	.67	.57	.46
P22P02	22	.75	.67	.63
P24P02	24	.79	.71	.66
P28P02	28	1.10	.93	.81
P40P02	40	1.25	1.07	.94

GOLD SOLDERTAIL STANDARD

	1-24	25-49	50-99
8STG	.30	.27	.24
14SG	.35	.32	.29
16STG	.38	.35	.32
18STG	.52	.47	.43
20STG	.60	.56	.52
22STG	.70	.63	.57
24STG	.70	.63	.57
28STG	1.10	1.00	.90
40STG	1.75	1.55	1.45

TIN SOLDERTAIL · LOW PROFILE

	1-24	25-49	50-99
8CS2	.25	.16	.15
14CS2	.25	.18	.16
16CS2	.25	.20	.18
18CS2	.29	.28	.27
20CS2	.34	.32	.30
22CS2	.37	.36	.35
24CS2	.38	.37	.36
28CS2	.45	.44	.43
40CS2	.63	.62	.61

3 LEVEL GOLD WIRE WRAP SOCKETS

Sockets purchased in multiples of 50 per type may be combined for best price.

	1-9	10-24	25-99	100-249	250-999
8 pin*	.40	.36	.34	.31	.27
14 pin*	.45	.39	.37	.34	.32
16 pin*	.50	.42	.40	.36	.34
18 pin	.70	.60	.55	.50	.45
20 pin	.90	.80	.75	.65	.62
22 pin*	.95	.85	.80	.70	.65
24 pin	.95	.85	.80	.70	.65
28 pin	1.25	1.15	1.00	.95	.90
40 pin	1.65	1.45	1.35	1.20	1.10

All sockets are GOLD 3 level closed entry *End and side stacable. 2 level, Solder Tail, Low Profile, Tin Sockets and Dip Plugs available. CALL FOR QUOTATION

unbeatable GREAT JUMPERS

FLAT RIBBON CABLE ASSEMBLIES AT AFFORDABLE PRICES

- Choice of 3 types of end connectors molded on and factory tested.
- Daisy chain and single-end also available.
- 5 popular sizes to choose from: 20, 26, 34, 40 and 50 contacts, each with line-by-line probe access holes.
- Choice of 2 cable types and 5 lengths.

FLAT RIBBON CABLE

Stranded, 28 AWG with laminated PVC insulation.

"Electric Pink" cable has red stripe on one edge for orientation. Used only on double-end and daisy chain assemblies.

"Rainbow" cable is coded in standard 10-color sequence on front. Serpentine striping on back aids in identifying wire number and wire group during tear-down separation for discrete wire terminations. Used only on single-end jumpers.

CARD-EDGE JUMPERS

No. Contacts	DOUBLE END Electric Pink 6"	SINGLE END Rainbow 36"	DAISY CHAIN (3 connectors) Electric Pink 6"
20	924052-06-R $5.69	924062-36-R $4.11	924092-06-R $8.42
26	924053-06-R $6.46	924063-36-R $4.88	924093-06-R $9.54
34	924054-06-R $7.96	924064-36-R $6.15	924094-06-R $11.74
40	924055-06-R $9.29	924065-36-R $7.20	924095-06-R $13.70
50	924056-06-R $10.10	924066-36-R $8.21	924096-06-R $14.86

PCB JUMPERS

No. Contacts	DOUBLE END Electric Pink 6"	SINGLE END Rainbow 36"
20	924032-06-R $2.57	924042-36-R $2.55
26	924033-06-R $3.32	924043-36-R $3.31
34	924034-06-R $3.95	924044-36-R $4.13
40	924035-06-R $4.57	924045-36-R $4.84
50	924036-06-R $5.62	924046-36-R $5.97

SOCKET JUMPERS

No. Contacts	DOUBLE END JUMPER ASSEMBLIES Electric Pink Cable 6"	18"	36"	SINGLE END Rainbow 36"	DAISY CHAIN (3 connectors) Electric Pink 6"
20	924002-06-R $3.70	924002-18-R $ 4.16	924002-36-R $ 4.85	924012-36-R $3.12	924072-06-R $ 5.44
26	924003-06-R $4.78	924003-18-R $ 5.38	924003-36-R $ 6.28	924013-36-R $4.04	924073-06-R $ 7.02
34	924004-06-R $6.25	924004-18-R $ 7.05	924004-36-R $ 8.25	924014-36-R $5.30	924074-06-R $ 9.18
40	924005-06-R $7.33	924005-18-R $ 8.27	924005-36-R $ 9.68	924015-36-R $6.22	924075-06-R $10.76
50	924006-06-R $9.15	924006-18-R $10.31	924006-36-R $12.05	924016-36-R $7.73	924076-06-R $13.43

double-row JUMPER HEADERS

Ideal mates for "GREAT JUMPERS"

- Solder to PC boards for instant plug-in access via socket-connector jumpers
- .025" square posts are molded into plastic header strip on a .10" x .10" matrix
- Choice of straight or right angle configurations

STRAIGHT

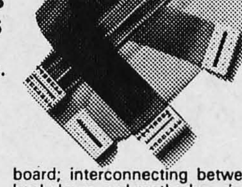
No. Posts	Dim. "A"	Dim. "B"	Part Number	Price 2 sets
20	1.0	0.9	923862-R	$ .98
26	1.3	1.2	923863-R	$ 1.28
34	1.7	1.6	923864-R	$ 1.64
40	2.0	1.9	923865-R	$ 1.94
50	2.5	2.4	923866-R	$ 2.36

RIGHT-ANGLE

No. Posts	Dim. "A"	Dim. "B"	Part Number	Price 2 sets
20	1.0	0.9	923872-R	$ 1.20
26	1.3	1.2	923873-R	$ 1.52
34	1.7	1.6	923874-R	$ 1.96
40	2.0	1.9	923875-R	$ 2.30
50	2.5	2.4	923876-R	$ 2.82

DIP JUMPERS

FLAT RIBBON CABLE ASSEMBLIES WITH DIP CONNECTORS



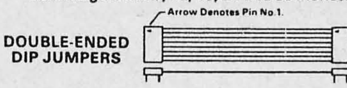


- Available with 14, 16, 24 and 40 contacts.
- Mate with standard IC sockets.
- Fully assembled and tested.
- Integral molded-on strain relief.
- Line-by-line probeability.

A P DIP Jumpers are the low-cost, high-quality solution for jumpering within a PC board; interconnecting between PC boards, backplanes and motherboards; interfacing Input/Output signals; and more.

All assemblies use rainbow cable. Standard lengths are 6, 12, 18, 24 and 36 inches.

WOW



DOUBLE-ENDED DIP JUMPERS

No. Contacts	Length 6"	Length 12"	Length 18"	Length 24"	Length 36"
14	924106-6-R $2.41	924106-12-R $2.61	924106-18-R $2.82	924106-24-R $3.02	924106-36-R $3.43
16	924116-6-R $2.65	924116-12-R $2.88	924116-18-R $3.11	924116-24-R $3.34	924116-36-R $3.80
24	924126-6-R $4.15	924126-12-R $4.50	924126-18-R $4.85	924126-24-R $5.20	924126-36-R $5.90
40	924136-6-R $6.93	924136-12-R $7.52	924136-18-R $8.11	924136-24-R $8.70	924136-36-R $9.88



PRIORITY ONE ELECTRONICS ©
16723B Roscoe Blvd. Sepulveda, CA 91343

Terms: Visa, MC, BAC, Check, Money Order, C.O.D. U.S. Funds Only. CA residents add 6% sales tax. Minimum order $10.00. Prepaid U.S. orders less than $75.00 include 5% shipping and handling, minimum $2.50. Excess refunded. Just in case . . . please include your phone no.

Prices subject to change without notice. We will do our best to maintain prices thru June 1979.

OEM and Institutional inquiries invited.

phone orders welcome (213) 894-8171, (800) 423-5633

We have the Best Prices on 2102's, 2114's, 4116's. We have 5V 2716's in Stock.

We make Custom Ribbon Cable Assemblies while you wait! We have the Best Prices on 2102's, 2114's, 4116's.

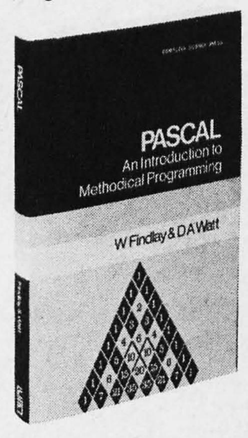
We make Custom Ribbon Cable Assemblies while you wait! We have the Best Prices on 2102's, 2114'2, 4116's.

TRS-80 Microcomputer Technical Reference Handbook

Radio Shack has published a technical reference handbook for their TRS-80 microcomputer system. The illustrated 108 page book is intended primarily for technically oriented persons with a good working knowledge of digital logic circuits. Written in a straightforward, informal manner, the *TRS-80 Microcomputer Technical Reference Handbook* includes technical information and schematic diagrams for both Level I and II TRS-80 systems. Topics covered in the book include: Theory of Operation, Adjustments and Troubleshooting, The Outside World (connections to control external devices), parts list and fold-out schematics. The handbook is priced at $9.95 and is available from Radio Shack stores and dealers. For further information, contact Radio Shack, 1400 One Tandy Center, Fort Worth TX 76102.

Circle 568 on inquiry card.

Pascal: An Introduction to Methodical Programming



This book, intended for use in a first course in programming, is based on the Pascal language. It assumes no prior knowledge of computing and only elementary mathematical skill. It emphasizes programming principles, good style, and a methodical approach to program development. This introduction to Pascal includes a thorough treatment of both the fundamental language features and the few features which are not truly fundamental. The programming technique of incremental refinement is imparted by consistent example throughout the book. In addition, two chapters are devoted exclusively to programming methodology. Each chapter is followed by a number of exercises, answers to some of which are provided. This 306 page book is priced at $10.95. It is published by Computer Science Press Inc, 9125 Fall River Ln, Potomac MD 20854.

Circle 569 on inquiry card.

Z-80 Instruction Handbook

The *Z80 Instruction Handbook* by Nat Wadsworth is a handy, compact reference providing a clear detailed explanation of the Z-80 microprocessor instruction set. Standard Zilog mnemonics are used throughout and machine codes are presented in both octal and hexadecimal format. An index lists all instructions alphabetically along with machine codes and timing information. This 128 page book is priced at $4.95. For further information, contact Scelbi Publications, POB 133 PP STN, Milford CT 06460.

Circle 570 on inquiry card.

Connect Your Computer to an Automatic Musical Instrument

Vestal Press has recently announced the release of its 15th catalog. It contains all types of automatic musical instruments including music boxes, carousel organs, orchestrions (mechanical orchestras), reproducing pianos (player pianos that play with full artistic fidelity), violin playing machines, and all sorts of unusual music devices. The catalog is available for $2 from Vestal Press, Dept B, POB 97, Vestal NY 13850.

Circle 571 on inquiry card.

Free Catalogs from Hayden Book Company

Two new catalogs that include personal computing and professional computing books are available from Hayden Book Co Inc. They feature Hayden's new and forthcoming books on introductory computing, programming, and applications and advanced technology. For free copies of either the personal or the professional computing catalog, write to Hayden Book Co Inc, 50 Essex St, Rochelle Park NJ 07662.

Circle 572 on inquiry card.

TRS-80 Monthly Newsletter

The *TRS-80 Monthly Newsletter* contains articles and programs (with complete program listings and instructions) related to business, personal finance, money management, games, practical applications and gambling. The programs are also available on cassette or floppy disk. A summary of the latest TRS-80 system developments and a list of TRS-80 related software are published in every issue. A one year subscription to this newsletter is $24. For more information, contact Mathematical Applications Service, POB 149 RS, New City NY 10956.

Circle 573 on inquiry card.

Comprehensive Microprocessor Design Manual Announced by TI



This self-teaching microprocessor design manual, written for both beginners and experts, is available from Texas Instruments Inc, Mail Station 54, POB 225012, Dallas TX 75222. *9900 Family Systems Design and Data Book* offers more than 1,000 pages of educational and applications information that can help users develop a deeper understanding of the complex technology and the potential in microprocessors.

The first chapter discusses the semiconductor technology advances on which today's microprocessors are based. It also provides guidelines for selection and application of microprocessors and microcomputers, and lists general and basic design decisions. The second chapter is a product selection guide, covering the complete TI 9900 family of microprocessors, microcomputers, peripheral support circuits, assembled microcomputer modules, software and development systems. Chapter Three moves step-by-step through a "first encounter" with a 9900, describing basic concepts in an introductory application. Chapters Four, Five, and Six cover hardware and software design, architecture and interfacing techniques, programming methods and the instruction set.

Chapter Seven contains reference materials for development systems used by experienced system designers to develop 9900 software programs, debug, and prototype final systems. It also includes in-depth technical specifications on all currently available products in the 9900 family.

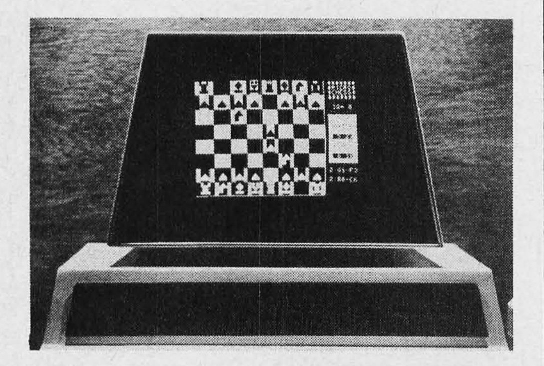
Chapter Nine offers detailed examples of real world uses of TI's 9900 product family in the design of a low cost data terminal, a floppy disk controller, and a simulated industrial control application.

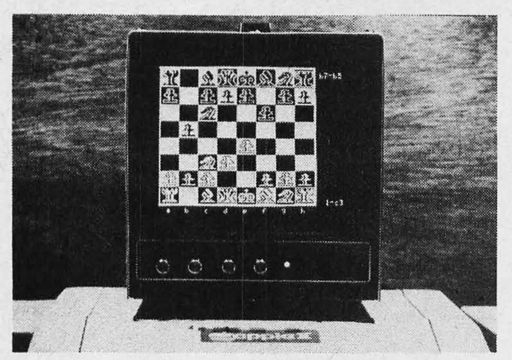
The *9900 Family Systems Design and Data Book* is priced at $9.95 (soft cover).

Circle 574 on inquiry card.

← Circle 312 on inquiry card.

Microchess for the PET and Apple Computers





Microchess 2.0, developed by Peter Jennings, has been designed for the 8 K PET and the 16 K Apple computers. In 6502 machine language, it offers 8 levels of play to suit everyone from the beginner to the serious player. At its highest level the program plays a good game and will beat most average players and many other chess playing programs. It examines positions by as many as six moves ahead, and includes a chess clock for tournament play. Microchess checks every move for legality, handles castling and en passant pawn captures, and displays the current position on a graphic chessboard. You can play white or black, set up and play from special board positions, or watch the computer play against itself. Microchess 2.0 is available for $19.95 from Personal Software, POB 136, Cambridge MA 02138.

Circle 609 on inquiry card.

New Publication on Patching and Programming from *Polyphony*

The Source is a compilation of analog music synthesizer patch charts which have appeared in *Polyphony* magazine plus some extras. The magazine and this book adhere to two important concepts: to show the average synthesist how to do it, and to promote and publish information exchange between synthesists. The first of the six chapters in the handbook is spent familiarizing the reader with the standard symbology used to represent various synthesizer modules. Another chapter contains patches which are most useful for actually playing tunes. In the techniques chapter is found a multitude of patches which provide insight into how to use modules efficiently and imaginatively. The final chapter on software contains the basis for an understanding of a few of today's newest musical tools. Program listings for computer music are provided. This 124 page book costs $4 and is available from Polyphony Publishing Co, 1020 W Wilshire Blvd, Oklahoma City OK 73116.

Circle 610 on inquiry card.

On Screen Text Editors for 8080 Systems

Two Daisy text editing programs allow fully interactive visible text editing and advanced word processing and formatting, using a serial video terminal. The editors provide extended file usage, and use dynamic screen imaging to minimize disruptive screen activity so that the editors can be used on a slow (2400 bps) video terminal.

WPDaisy is the word processing version of this system which includes both space and proportional justification. WPDaisy allows calling disk files while formatting, and has 26 in-memory buffers. Also included is a mail merge program which is useful in producing form letters and labels.

The TSA/os version is $125 for Daisy, and $300 for WPDaisy. The CP/M version is $175 for Daisy; $350 for WPDaisy. For further information, contact TSA Software Inc, 39 Williams Dr, Monroe CT 06468.

Circle 611 on inquiry card.

Multi-Universal Integrated Circuit Plug-In Adapter

This multi-universal integrated circuit plug-in adapter accepts integrated circuit patterns of up to 40 leads, including large scale integration, medium scale integration, and programmable memory devices. The adapter is used for mounting a variety of mixed devices which will then plug into any standard universal integrated circuit packaging panel.

The P/N 640-MUI adapter will accept the following packages: single-in-line, dual-in-line, memory package, interface, and large scale devices. The adapters are fabricated of 0.062 inch (0.157 cm) thick epoxy with electro-tin-plated circuitry. The plug-in contacts are brass, tin plated, with a gold plated berrylium copper four-tine spring socket member.

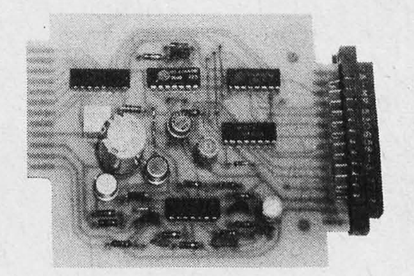
They are available at prices ranging from $2 to $6 per unit. For more information, contact Garry Manufacturing Co, 1010 Jersey Av, New Brunswick NJ 08902.

Circle 612 on inquiry card.

New Family of RS-232 Switching Units

A new family of low cost miniature switching units has been introduced by Giltronix Inc, 3156 Avalon, Palo Alto CA 94306. The family, called RS232-X, switches serial RS-232 peripherals between several driving sources. Model RS232-X3 allows three driving sources. By turning the three position switch mounted on the RS232-X3, the user can select the driving device that will exchange data with the peripheral unit. A unique arrangement allows the cascading of two or more RS232-X switches, thereby expanding the selection from three devices to five or more. Model RS232-XF is similar to the RS232-X3 but switches additional signals. Both come with 25 pin female connectors. The price of the RS232-X3 is $64.95 assembled, and $47.95 in kit form. The RS232-XF is $78.95 assembled and $59.95 in kit form.

Circle 613 on inquiry card.



Analog Interface Card

The ADAK-1 board is a general purpose analog interface for 8 bit microcomputers. It includes a monolithic digital to analog converter, a 5 pole low pass filter for waveform generation, an audio amplifier, an eight channel analog input multiplexor, and a comparator to perform analog to digital conversions by successive approximation routines. This combination permits real time music generation by Fourier synthesis techniques and permits the outputs from up to eight joystick channels, thermometers, light sensors, or other devices to be digitized. A modified version, ADAK-1 PET, plugs directly into Commodore PET computers. The software cassette supplied with this version includes programs for machine language coding, music generation, Fourier waveform synthesis, analog to digital conversion and several paddle input games. Both versions are completely tested and include connectors and instructions. ADAK-1 is priced at $69.50 and the ADAK-1 PET version is $99.50. For further information, contact Technical Hardware Inc, POB 3609, Fullerton CA 92634.

Circle 614 on inquiry card.

*MEMORY SALE! have it your way . . .

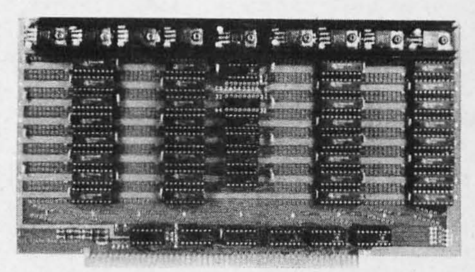
16K $295.00!! (4MHz) (Reg. $370.00)   32K $485.00!! (4MHz) (Reg. $620.00)

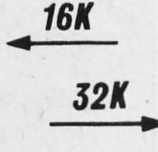
ALL BOARDS ASSEMBLED AND TESTED . . . . . . (KIT PRICING AVAILABLE)

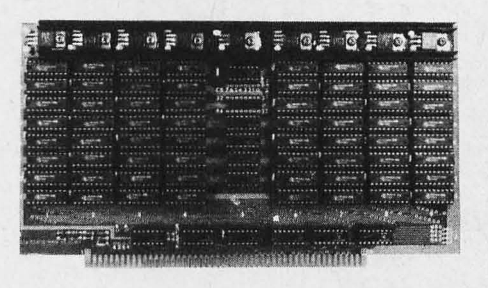
- Extended addressing allows board to exist anywhere in 256K memory on standard S-100 bus
- LOW Power, 1.6 amp per 16K
- 9 Regulators for perfect heat distribution
- Static, of course
- Phantom line
- Each 4K block locateable anywhere
- Fully tested and burned in for 48 hours

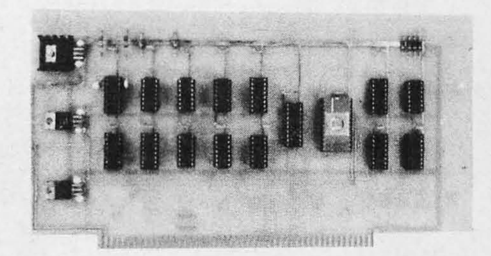
ADD-ON MEMORY CHIPS – $4.95 EACH!! (TMS 4044 or MM 5257) – *8 Chips – Minimum Order*

*Sale extended one last month



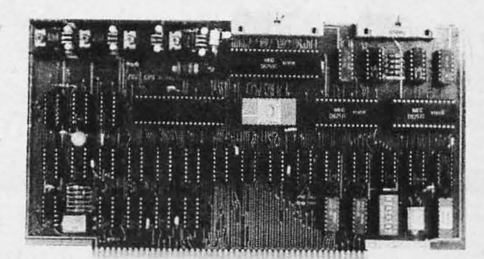






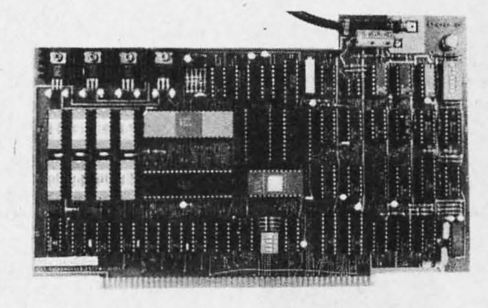
Z-80 CPU (one serial chip set, less eprom) $195.00 (Reg. $280.00)

- 2 Parallel + 2 Serial Port
- 2 MHz or 4 MHz Switch Selectable
- Baud Rates 150-9600
- Power on Jump to On/Board Eprom (2708 or 2716)
- Memory Management on A16 and A17



VIDEO TERMINAL SIMULATOR $295.00 (Reg. $400.00)

- Plugs into S-100 Bus and simulates all functions of a Soroc or other RS-232 type terminal. A simple video monitor such as a Sanyo or Sony TV will perform as a smart terminal by writing into an IO Port.
- 2K Eprom, 4K Ram (2 video pages on 16 x 64)
- Lower Case Descenders (16 x 64 or 24 x 80)
- Tabs, protected fields, home/load cursor, blink, reverse video, underline, page erase, etc. (Intel 8275 CRT controller)



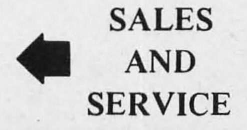
DOUBLE DENSITY DISC CONTROLLER $385.00

- CPM* Compatible, TARBELL Pin-out compatible
- On/Board Boot
- 2 or 4 MHz Jumper Selectable
- 8 inch Single or Double sided (5.25 inch available)
- 52 Sectors, runs CPM*, IBM Format

*CPM is a trademark of Digital Research, Inc.

West:
DELTA PRODUCTS
1653 E. 28th Street
Long Beach, Calif 90806
Tel (213) 595-7505

ORDER NOW ! !



East:
DELTA PRODUCTS
1254 South Cedar Road
New Lenox, Illinois 60451
Tel (815) 485-9072

Circle 89 on inquiry card.

Dealer inquiries invited.

Personal checks must clear before shipment • 90 day unconditional warranty • B of A, M/C Okay

PLACE ORDERS TOLL FREE:
800/421-5809 Continental U.S.
800/262-1710 Inside California

LEEDEX MONITOR
- 12" Black and White
- 12MHZ Bandwidth
- Handsome Plastic Case

$139.00



$179

KIM-1 Module monitor, programs stored in 2048 ROM Bytes, User Manual, Wall size schematic, Hardware manual, Programming manual, Programmers reference card, and Keyboard display.

CASES — $29.95

THE SYM-1 $245

6502 — based single board computer with keyboard/display, KIM-1 hardware compatible, complete documentation.

SYM—1 CASE $39.95

MICROPROCESSORS	
F8	$16.95
Z80 (2MHz)	$20.00
Z80A (4MHz)	$14.95
CDP1802CD	$17.95
6502	$11.95
6800	$16,95
6802	$25.00
8008-1	$12.95
8035	$20.00
8035-8	$21.00
8080A	$10.00
8085	$23.00
TMS9900TL	$49.95
8080A SUPPORT DEVICES	
8212	$ 2.90
8214	$ 4.65
8216	$ 2.75
8224 (2MHz)	$ 4.30
8224-4 (4MHz)	$ 9.95
8226	$ 2.75
8228	$ 6.40
8238	$ 6.40
8243	$ 8.00
8251	$ 7.50
8253	$20.00
8255	$ 6.45
8257	$20.00
8259	$20.00
8275	$75.00
8279	$18.50
USRT	
S2350	$10.95
UARTS	
AY5-1013A	$ 5.25
AY5-1014A	$ 8.25
TR1602B	$ 5.25
TMS6011	$ 5.95
IM6403	$ 9.00
BAUD RATE GENERATORS	
MC14411	$10.00
14411 Crystal	$ 4.95
6800 PRODUCT	
6810P	$ 4.00
6820P	$ 6.60
6821P	$ 6.60
6828P	$11.25
6834P	$16.95
6850P	$ 8.65
6852P	$11.00
6860P	$ 9.25
6862P	$12.00
6871P	$28.75
6875P	$ 8.75
6880P	$ 2.50
CHARACTER GENERATORS	
2513 Upper (-12+5)	$ 6.75
2513 Lower (-12+5)	$ 6.75
2513 Upper (5 volt)	$ 9.75
2513 Lower (5 volt)	$10.95
MCM6571 up scan	$10.95
MCM6571A down scn	$10.95
PROMS	
1702A	$5.00
2708	$ 9.95
2716 (5+12) TI	$60.00
2716 (5v) INTEL	$60.00
2758 (5v)	$30.00
DYNAMIC RAMS	
416D/4116 (200ns)	$12.50
2104/4096	$ 4.00
2107B-4	$ 3.95
TMS4027/4096 (300ns)	$ 4.00

STATIC RAMS	1—15	16—100
21L02 (450ns)	$ 1.50	$ 1.20
21L02 (250ns)	$ 1.75	$ 1.50
2101-1	$ 2.95	$ 2.60
2111-1	$ 3.25	$ 3.00
2112-1	$ 2.95	$ 2.65

FLOPPY DISK CONTROLLERS	
1771B01	$39.95
1791	$49.95
KEYBOARD CHIPS	
AY5-2376	$13.75
AY5-3600	$13.75
MM5740	$18.00

JADE'S *DOUBLE DENSITY Controller Board*

KIT: $249.00
Assmb. & Tstd: $299.00

- Single or Double Density Recording
- Full Size or Mini Floppy
- CP/M Compatable in either density
- Programmed Data Transfer, no DMA
- Controls up to 8 drives
- IBM format in either density
- Software Selectable Density

■ This controler utilizes the proven reliability of the IBM standard format as well as the lastest phase-locked-loop for data seperation. ■ All clocks are generated from an on-board cyrstal oscillator ■ Right precompensation is used to enhanced data recovery reliability in the double density mode ■ Density selection is enitirely transparent to the user ■ Single and double density diskettes can be mixed on the same system.

NEW!
JADE Z80 BOARD
IMPROVED DESIGN AND FEATURES

- ON BOARD 2708 or 2716 EPROM
- VERY RELIABLE AT 4 MHZ OR 2 MHZ
- POWER ON JUMP AND RESET
- ON BOARD USART (8251)

2MHz	
Kit	$135.00
Assembled & Tested	$185.00
4 MHz	
Kit	$149.95
Assembled & Tested	$199.95
Bare Board	$ 35.00

JADE 8080A
with full documentation

Kit	$100.00
Assembled & Tested	$149.95
Bare Board	$ 30.00

EPROM BOARD KITS
JG8/16 (uses 2708 or 2716) **$69.95**

APPLE•TRS-80•APPLE•TRS-80•APPLE•TRS-80

Jade Memory Expansion Kits for TRS-80 and Apple!
4116's

Everything a person needs to add 16K of memory. Chips come neatly packaged with easy to follow directions. In minutes your machine is ready for games and more advanced software.

$82.00

"IMSAI"-TYPE CARD GUIDE SPECIAL:
Regular Price 30¢ each

SPECIAL: 10 for $1.00!

Rockwell AIM-65: The Head-Start in Microcomputers

A KIM-1 compatible machine with on-board printer and a real keyboard!

$375.00 w/1K RAM
$450.00 w/4K RAM

4K assembler/editor in ROM:	$ 80.00
8K BASIC in ROM:	$100.00
Power supply:	$ 59.95
Case for AIM-65:	$ 49.95

Special Package Price: $599.00
AIM-65 (4K), Power Supply, Case, and 8K BASIC ROM

New Prices

DYNAMIC RAM BOARDS EXPANDABLE TO 64K

32K VERSION • KITS

Uses 4115 (8Kx1, 250ns) Dynamic RAM's, can be expanded in 8K increments up to 32K:

8K	$159.00	24K	$249.00
16K	$199.00	32K	$299.00

4115 SALE
8 for $39.95

64K VERSION • KITS

Uses 4116 (16Kx1, 200ns) Dynamic RAM's, can be expanded in 16K increments up to 64K:

16K	$249.00	48K	$469.00
32K	$369.00	64K	$569.00

DYNABYTE *Naked Terminal*
$350.00

FEATURES & BENEFITS
■ 80 character by 24 line format (Model 57) ■ Completely self contained terminal electronics, just add CCTV monitor and key board ■ No support software required ■ Switch selectable modes: Half Duplex, Full Duplex, Block mode ■ Block mode allows for editing before transmit ■ Keyboard interface provided ■ Fully assembled, socketed, tested, burned in, and guaranteed for a full year from date of purchase ■ Video is switch selectable "Black-on-White" or "White-on-Black"

TEXTOOL ZIP* DIP® II Sockets

***ZERO INSERTION FORCE** sockets*

PRICES:	
16 pin Zip Dip II	$5.50
24 pin Zip Dip II	$7.50
40 pin Zip Dip II	$10.25

★ STATIC RAM SPECIALS ★

2114's, low power (1024x4)	1-15	16-99	100 +
450ns	8.00	6.95	5.50
300ns	9.00	8.00	6.50
TMS4044/MM5257, low power			
450ns	8.00	7.50	6.50
300ns	9.95	8.75	8.00
4200A (4Kx1, 200ns)	9.95	8.50	8.00
410D (4K x 1, 200 ns)	8.25	7.00	6.75

S-100 EDGE CONNECTORS

Soldertail	$3.25 each
10 for $35.00	
Wire Wrap	$4.00 each
10 for $40.00	

STATIC RAM BOARDS

JADE 8K

Kits:	450ns	$125.95
	250ns	$149.75
Assembled & Tested:		
	450ns	$139.75
	250ns	$169.75
Bare Board:		$ 25.00

16K — Uses 2114's (low power)
Assembled & Tested:

RAM 16 (250ns)	$375.00
RAM 16B (450ns)	$325.00

16K with memory management
Assembled & Tested:

RAM 65 (250ns)	$390.00
RAM 65B (450ns)	$350.00

32K Static
Assembled & Tested:

250ns	$795.00
450ns	$725.00
250ns Kit	$575.00

JADE Computer Products

4901 W. ROSECRANS AVENUE
Department "F" 3
HAWTHORNE, CALIFORNIA 90250
U.S.A.

Telephone:
(213) 679-3313
(800) 421-5809 Continental U.S.
(800) 262-1710 Inside California








Cash, checks, money orders, and credit cards accepted. Add freight charge of $2.50 for orders under 10 lbs. and $1.00 service charge for orders under $10.00. Add 6% sales tax on all parts delivered in California. Discounts available at OEM quantities.

WRITE FOR OUR FREE CATALOG
All prices subject to change without notice.

Circle 195 on inquiry card.

Full Standard PILOT on PET

Commodore PET owners can get full standard PILOT on a minimum size PET with the PETPILOT language processor and editor which is suitable for preparing long programs of up to 80,000 characters. The product features full BASIC in compute statements as well as two new keywords designed to make PILOT programming easier and faster. All language features of the most recent PILOT standard are implemented. Only the tape drive supplied with the PET is required to run any PILOT program. While simple PILOT programs can be created on a single drive PET, authors writing long programs will need the second cassette drive offered by Commodore.

The package offered by the PET-PILOT project contains both programs, a sample PILOT program, a teacher's manual, a quick reference card, and licenses to run the programs on a single PET. A tutorial course of 4 one hour lessons in effective use of PILOT is also available. The basic package costs $12, and the tutorial is an extra $8. Both products can be ordered by specifying the PET serial number to be licensed. For further information contact Dave Gomberg, 7 Gateview Ct, San Francisco CA 94116.

Circle 620 on inquiry card.

FORTRAN Compiler for 6800 Produces Relocatable Object Code

A FORTRAN compiler for 6800 microprocessors, which produces relocatable object code in a Motorola compatible format, has been introduced by Smoke Signal Broadcasting, 31336 Via Colinas, Westlake Village CA 91361. Programs are compiled to run under the company's DOS-68 disk operating system for scientific applications, number crunching and multidimensional array processing. The compiler is also complemented by Smoke Signal's Linking Loader for loading the object listing into any portion of memory specified.

Requiring 24 K bytes of user programmable memory, the compiler has a data initialization capability, features arithmetic and logical IF statements, and handles sequential access files so that up to four files can be opened at any one time. FORTRAN library subroutines can also be built.

The 6800 FORTRAN compiler is priced at $99.

Circle 621 on inquiry card.

KIM-1 Control System

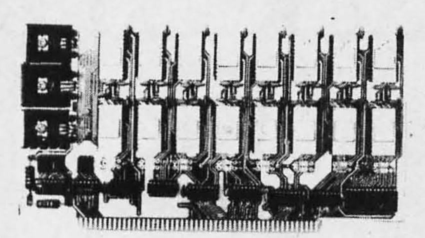
PCROS is a Process Control Real-time Operating System for the KIM-1 microcomputer board. The operating system is designed to function in the 1 K byte KIM-1 programmable memory. PCROS can control up to twelve switches and is driven by a real time clock routine that makes use of the KIM-1 timer and interrupt circuitry.

A process control language interpreter has been included as an integral part of PCROS. The interpreter utilizes the on board KIM-1 keyboard and diplay. The process control language interpreter provides nine commands for application program development: set switches, hold full-second current settings (up to 255), hold quarter-second current settings (up to 255), repeat command sequence, reset repeat loop, go to subroutine, return from subroutine, load and execute next program (from cassette tape), and halt. Application programs can contain up to 56 commands.

PCROS on KIM format cassettes with users manual is priced at $14.95. The assembly listing is available for $24.95. For further information contact H Geller Computer Systems, POB 350, New York NY 10040.

Circle 622 on inquiry card.

16K EPROM CARD-S 100 BUSS



$59.95 KIT

OUR BEST SELLING KIT!

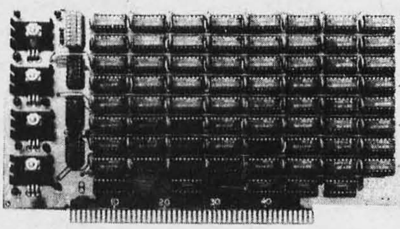
USES 2708's!

Thousands of personal and business systems around the world use this board with complete satisfaction. Puts 16K of software on line at **ALL TIMES!** Kit features a top quality soldermasked and silk-screened PC board and first run parts and sockets. All parts (except 2708's) are included. Any number of EPROM locations may be disabled to avoid any memory conflicts. Fully buffered and has WAIT STATE capabilities.

OUR 450NS 2708'S ARE $8.95 EA. WITH PURCHASE OF KIT

ASSEMBLED AND FULLY TESTED ADD $25

8K LOW POWER RAM KIT-S 100 BUSS
250 NS SALE!



ADD $5 FOR 250NS!

$129 KIT

(450 NS RAMS!)

Thousands of computer systems rely on this rugged, work horse, RAM board. Designed for error-free, NO HASSLE, systems use.

KIT FEATURES:

1. Doubled sided PC Board with solder mask and silk screen layout. Gold plated contact fingers.
2. All sockets included.
3. Fully buffered on all address and data lines.
4. Phantom is jumper selectable to pin 67.
5. FOUR 7805 regulators are provided on card.

Blank PC Board w/Documentation **$29.95**
Low Profile Socket Set...**13.50**
Support IC's (TTL & Regulators) **$9.75**
Bypass CAP's (Disc & Tantalums) **$4.50**

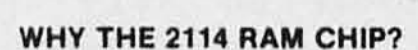
ASSEMBLED AND FULLY BURNED IN ADD $30

16K STATIC RAM KIT-S 100 BUSS

$295 KIT



FULLY STATIC, AT DYNAMIC PRICES



WHY THE 2114 RAM CHIP?

We feel the 2114 will be the next industry standard RAM chip (like the 2102 was). This means price, availability, and quality will all be good! Next, the 2114 is FULLY STATIC! We feel this is the **ONLY** way to go on the S-100 Buss! We've all heard the HORROR stories about some Dynamic Ram Boards having trouble with DMA and FLOPPY DISC DRIVES. Who needs these kinds of problems? And finally, even among other 4K Static RAM's the 2114 stands out! Not all 4K static Rams are created equal! Some of the other 4K's have **clocked** chip enable lines and various timing windows **just as critical** as Dynamic RAM's. Some of our competitor's 16K boards use these "tricky" devices. But not us! The 2114 is the ONLY logical choice for a trouble-free, straightforward design.

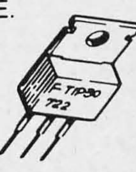
KIT FEATURES:

1. Addressable as four separate 4K Blocks.
2. ON BOARD BANK SELECT circuitry. (Cromemco Standard!). Allows up to 512K on line!
3. Uses 2114 (450NS) 4K Static Rams.
4. ON BOARD SELECTABLE WAIT STATES.
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers.
6. All address and data lines fully buffered.
7. Kit includes ALL parts and sockets.
8. PHANTOM is jumpered to PIN 67.
9. LOW POWER: under 2 amps TYPICAL from the +8 Volt Buss.
10. Blank PC Board can be populated as any multiple of 4K.

BLANK PC BOARD W/DATA—**$33**
LOW PROFILE SOCKET SET—**$12**
SUPPORT IC'S & CAPS—**$19.95**
ASSEMBLED & TESTED—**ADD $30**
2114 RAM'S—**8 FOR $69.95**

COMPLEMENTARY POWER TRANSISTORS

SILICON NPN AND PNP. TO-220 CASE.
VCEO - 40V PD - 30 WATTS
FOR AUDIO POWER AMPS, ETC.



TIP29 - NPN
TIP30 - PNP

YOUR CHOICE
3 FOR $1

16K DYNAMIC RAM CHIP

16K X 1 Bits. 16 Pin Package. Same as Mostek 4116-4. 250 NS access. 410 NS cycle time. Our best price yet for this state of the art RAM. 32K and 64K RAM boards using this chip are readily available. These are new, fully guaranteed devices by a major mfg. **VERY LIMITED STOCK!**

8 FOR $89.95

NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.

450 NS!

2708 EPROMS

Now full speed! Prime new units from a major U.S. Mfg. 450 N.S. Access time. 1K x 8. Equiv. to 4-1702 A's in one package.

~~$15.75 ea.~~ **$9.95** ~~4 FOR $50.00~~

PRICE CUT

NATIONAL SEMICONDUCTOR ***NEW!***

CAR CLOCK MODULE - #MA6008

$6.99 each

Originally used by HYGAIN to indicate time and channel on an expensive C.B. Mini size, self contained module. Not a Kit. Four digits plus flashing indicator for seconds. Includes MM5369 and 3.58 MHZ crystal for super accurate time base. With hookup data.

INCLUDES CRYSTAL TIMEBASE! WORKS ON 12 VDC!

MFGR's CLOSEOUT LIMITED QTY.

EXPERIMENTER'S HEATING PLATE

Large Manufacturers Surplus. 5¼x10½ in. Made of 3/8 in. tempered glass with heating element laminated on back. Works off 120 VAC. Protected by thermostat and two thermal fuses. Rated 120 Watts. Use for any heating applications. Perfect for heating ferric chloride to increase PC Board etching efficiency. Units are brand new, non-submersible.

WHILE THEY LAST—$2.99 each

MALLORY COMPUTER GRADE CAPACITOR

30,000 MFD 15 WVDC
Small: 3 x 2 Inches
$1.99 ea. 3 For $4.99

New! REAL TIME Computer Clock Chip

N.S. MM5313. Features BOTH 7 segment and BCD outputs. 28 Pin DIP. **$4.95 with Data**

Z-80 PROGRAMMING MANUAL

By MOSTEK, or ZILOG. The most detailed explanation ever on the working of the Z-80 CPU CHIPS. At least one full page on each of the 158 Z-80 instructions. A MUST reference manual for any user of the Z-80. 300 pages. Just off the press. **$12.95**

GENERAL INSTRUMENT

FULL WAVE BRIDGE
4 AMP 600 PIV

LOOK!

3/4 IN. SQUARE - WITH LUGS

75¢ ea. 3 FOR $2 #LM-1

"THE COLOSSUS"
FAIRCHILD SUPER JUMBO LED READOUT

A full .80 inch character. The biggest readout we have ever sold! Super efficient. Compare at up to $2.95 each from others!

YOUR CHOICE
FND 843 Common Anode
FND 850 Common Cathode **$1.49 ea** **(6 for $6.95)**

Digital Research Corporation
(OF TEXAS)

P.O. BOX 401247Y GARLAND, TEXAS 75040 • (214) 271-2461

TERMS: Add 30¢ postage, we pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa, MasterCharge, and American Express cards. Tex. Res. add 5% Tax. Foreign orders (except Canada add 20% P & H. 90 Day Money Back Guarantee on all items.

Circle 100 on inquiry card.

The COMPUTER FACTORY

TO ORDER CALL TOLL FREE 800-223-7318



COMMODORE "PET"

only **$795** USED $549

- 14K ROM Operating system
- 8K RAM Memory
- 9" Video Monitor
- Built in Keyboard
- Digitally controlled tape

Today's best value in personal computers, along with the latest in peripherals, are always in stock at the COMPUTER FACTORY.

NEW! PET BUSINESS SYSTEM

The PET is now a truly sophisticated Business System with the Floppy Disk and Printer which makes an ideal cost efficient business system for most professional and specialized fields: medicine, law, research, engineering, education, etc.



PET 2001 -
16K ~~$995~~
32K $1295

- 16 or 32K Bytes Dynamic RAM • 14K ROM Operating System
- Upper/Lower Case and Graphics
- Full Sized Business Keyboard
- Full Screen Editing
- Operating system will support multiple Languages (BASIC resident)
- Machine Language Monitor
- 8K ROM Expansion Sockets
- 9" CRT

PERIPHERALS FOR PET

- 24K Memory Expansion $499
- 16K Memory Expansion 399
- PET to RS232 Serial 169
- 2 Way Serial/Communication 229
- Modem Board for PET 375
- Analog to Digital Board For 16 Devices............. 275
- Second Cassette Drive 95
- Parallel Printer Interface ... 199



PET PRINTERS
2023 **$849**
2022 **$995**
(Includes Tractors)

- Bottom and Rear Tractor Feed
- 8½" Paper Width
- 6504 Microprocessor Controlled
- 150 cps Bi-directional
- 4K ROM • ½K RAM
- Upper/Lower Case and Graphics
- 7 x 6 Dot Matrix



PET MUSIC BOX
Add music and sound effects to your programs. Compose, play, and hear music on your pet. Completely self-contained (no wiring). Free 3 programs including Star Wars theme, sound effects, etc. $39.

DUAL DRIVE FLOPPY DISK 2040 $1295

- 360 K Bytes Storage
- High Speed Data Transfer
- Plugs into IEEE Port
- 6504 Microprocessor
- 8K ROM Operating System
- 8K ROM Encoding and Decoding
- 4K RAM
- Uses Single or Dual sided Diskettes

SINGLE DRIVE FLOPPY UNIT 2041 $895

- Intelligent mini floppy
- 171.5K net user storage

SALE

Come In Or Call For Special Low Prices

WE'RE MOVING TO LARGER QUARTERS AND MUST SELL EVERYTHING IN OUR INVENTORY

FREE $35 of Software with purchase of any computer on this page



$1495 Complete!
16K Model add $200
32K Model add $500

Compucolor II

COMPUCOLOR II Disk-Based Model 3
Advanced hardware and software technology gives you:

- 13" Color Display
- Advanced Color Graphics
- 51K Disk Built-In
- 16K ROM Operating System
- 8K RAM User Memory
- 4K RAM Refresh
- 8080A Microcomputer
- RS-232 I/O

Every unit comes with an extended DISK-BASIC that has full file management capability resident in the COMPUCOLOR II in 16K of ROM. Color is fantastic, but COMPUCOLOR II has the power to handle complex tasks and small business applications. An impressive software library supplements your own creativity.

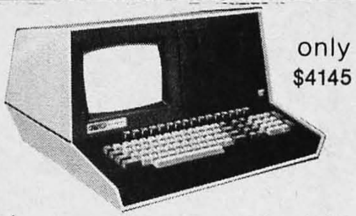
BUSINESS COMPUTER IMSAI

The low cost solution for all small business problems. A wide variety of software is available for all your needs.
PCS series include dual floppies, 32K RAM, I/O, DOS, BASIC

- PCS-42 (400KB) $3295
- PCS-44 (780KB) $3995

VDP-42 series adds video terminal, keyboard and VIO to above

- VDP-42 $4995
- VDP-44 $5595
- VDP/80 $7995



only $4145

BILLINGS MICROSYSTEM EXPANDABLE TO 2 MEGABYTES NEW!

features:

- 12" CRT
- 64K RAM
- Z-80
- Dual mini floppy
- 320 K Storage
- Graphics
- 94 Keys
- 2 RS-232 ports
- 1 parallel port

BUSINESS SOFTWARE AVAILABLE!



apple

APPLE 11
APPLE is a completely contained computer system, with BASIC in ROM, full ASC 11 keyboard in a lightweight molded carrying case. Expandable to 48K

APPLE 11 Disk	$595
APPLESOFT in ROM	$200
Interface Cards	$179

MICROPRODUCTS

• EPROM Prgmr	$ 99.95
• EPROM Socket Adpt	$ 14.95
• APPLR 11 Modem	$129.95

SUPER SOFTWARE

Word Processing For PET — This super advanced full function program will allow you to create text from PET or terminal keyboard. • INSERT • DELETE • CENTER • UNDERLINE • FULL SCREEN EDITING • MOVE LINES OR BLOCKS • SAVE TEXT ON TAPE • AUTO PRINT FEATURE....$45

Apple's Moving Average Plot Program — This fantastic program disk allows the statistician, mathematician, trader in stocks, money or commodities, the ability to maintain 30 database series of up to 300 values and plot 3 different moving averages of a series at the same time, in 3 different colors. Files can be updated, deleted, changed, extended, etc.
A sure value disk at only $40!
Word Processing For Apple on disk...$50

SORCERER

by EXIDY

- Z-80 Processor
- Keyboard and Numeric Pad
- 8K RAM (up to 32K)
- Serial and Parallel I/O
- Dual Cassette I/O
- 30 x 64 Display
- S-100 Compatible
- 64 User Defined Characters
- 240 x 512 Resolution Graphics

$895

Modular design allows use of both cassettes for programs and ROM PAC cartridges for future language introductions like APL, COBOL, PILOT, FORTRAN, etc. Add monitor and tape or disk for complete computer system.

SPECIAL
12" Video Monitor for SORCERER ($299 value)
ONLY
$125 with 8K unit
95 with 16K unit
65 with 32K unit

ANDERSON JACOBSON



841 I/O Terminal
Ideal for word processing and small businesses.

- ASC 11 Code
- 15 cps Printout
- High Quality Selective Printing
- Use Keyboard for PET
- Reliable heavy duty Mechanism
- Completely Refurbished by A.J.
- Service in 15 Major Cities

Parallel **$1095**
Serial **$1195**

RADIO SHACK • PET • SORCERER • APPLE • COMPUCOLOR • ETC.

PRINTERS • PRINTERS • PRINTERS

The COMPUTER FACTORY'S extensive inventory and wide selection of computer printers assures you of finding the printer best suited for your needs and specifications. The following printers work well with all known personal computers.

CENTRONICS	$1095
AXIOM (Parallel)	445
AXIOM (Serial)	520
TRENDCOM	375
INTEGRAL DATA	795
QUME or DIABLO	3400



Min Credit Card Order $75

VISA

Open Mon.-Fri. 10-6 Sat. 10-4

N.Y. residents add 8% sales tax • Same day shipment on prepaid and credit card orders • Add $5 shipping for computers, $3 for boards, $.25 each cassette tape.



INTERTEC's INTERTUBE video terminal
Full screen editing 24 x 80 display and microprocessor controlled.
Only $870

SPECIAL SALE
Buy any computer and take 10% off price of printer of your choice

TO ORDER CALL TOLL FREE 800-223-7318

The COMPUTER FACTORY 485 Lexington Avenue (212) 687-5001 750 Third Avenue (46th St.) (212) PET-2001 New York, N.Y. 10017 Foreign order desk - Telex 640055



Circle 27 on inquiry card.



Circle 19 on inquiry card.

The DATA-TRANS 1000

A completely refurbished IBM Selectric Terminal with built-in ASCII Interface.

Features:

- 300 Baud
- 14.9 characters per second printout
- Reliable heavy duty Selectric mechanism
- RS-232C Interface
- Documentation included
- 60 day warranty - parts and labor
- High quality Selectric printing Off-line use as typewriter
- Optional tractor feed available
- 15 inch carriage width

HOW TO ORDER DATA-TRANS 1000

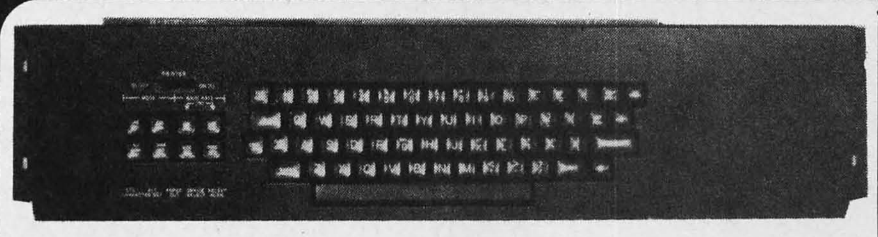
1. We accept Visa, Master Charge. Make cashiers checks or personal check payable to:

DATA-TRANS

2. All orders are shipped F.O.B. San Jose, CA
3. Deliveries are immediate

For orders and information

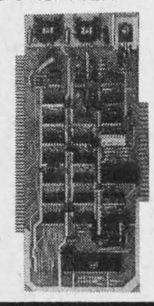
DATA-TRANS
2154 O'Toole St.
Unit E
San Jose, CA 95131
Phone: (408) 263-9246



ASCII KEYBOARD . . . . Mounted to DECWRITER Panel
By Cherry Products . . . ASSEMBLED 89.95

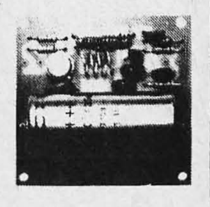
TIDMA⁂

• Tape Interface Direct Memory Access • Record and play programs without bootstrap loader (no prom) has FSK encoder/decoder for direct connections to low cost recorder at 1200 baud rate, and direct connections for inputs and outputs to a digital recorder at any baud rate • S-100 bus compatible • Board only $35.00 Part No. 112, with parts $110 Part No. 112A

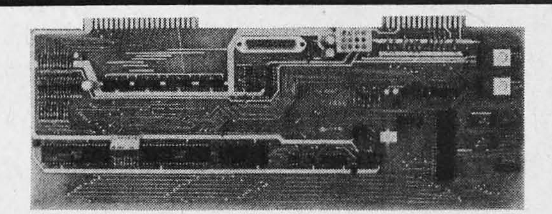


T.V. INTERFACE

• Converts video to AM modulated RF, Channels 2 or 3. So powerful almost no tuning is required. On board regulated power supply makes this extremely stable. Rated very highly in Doctor Dobbs' Journal. Recommended by Apple • Power required is 12 volts AC C.T., or +5 volts DC • Board only $7.60 part No. 107, with parts $13.50 Part No. 107A



(Illegal where prohibited by law.)



ASCII to Correspondence code converter

This bidirectional board is a direct replacement for the board inside the Trendata 1000 terminal. The on board connector provides RS-232 serial in and out. Sold only as an assembled and tested unit for $330.00. Part No. TA 1000C



apple II
With 16K $1024.00

To Order: Mention part number, description, and price. In USA, shipping paid for orders accompanied by check, money order, or Master Charge, BankAmericard, or VISA number, expiration date and signature. Shipping charges added to C.O.D. orders. California residents add 6.5% for tax. Outside USA add 10% for air mail postage and handling, no C.O.D.'s. Checks and money orders must be payable in US dollars. Parts kits include sockets for all ICs, components, and circuit board. Documentation is included with all products. Prices are in US dollars. No open accounts. To eliminate tariff in Canada boxes are marked "Computer Parts." Dealer inquiries invited. 24 Hour Order Line: (408) 226-4064

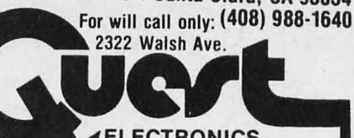
VISA · master charge

⁂ Circuits designed by John Bell

For free catalog including parts lists and schematics, send a self-addressed stamped envelope.

ELECTRONIC SYSTEMS Dept. B, P. O. Box 21638, San Jose, CA USA 95151

Circle 126 on inquiry card.



Same day shipment. First line parts only. Factory tested. Guaranteed money back. Quality IC's and other components at factory prices.

INTEGRATED CIRCUITS

7400TTL	
7400N	.17
7402N	.17
7404N	.19
7409N	.23
7410N	.17
7414N	.63
7420N	.17
7422N	1.39
7430N	.20
7442N	.50
7445N	.69
7447N	.60
7448N	.69
7450N	.17
7474N	.29
7475N	.49
7485N	.88
7489N	2.00
7490N	.43
7492N	.43
7493N	.43
7495N	.69
74100N	.90
74107N	.29
74121N	.34
74123N	.59
74125N	.39
74145N	.69
74150N	.95
74151N	.69
74154N	1.00
74157N	.69
74161N	.87
74162N	.87
74163N	.87
74174N	.96
74175N	.90
74190N	1.15
74192N	.87
74193N	.85
74221N	1.55
74298N	1.65
74365N	.95
74366N	.95
74367N	.95

74LS00 TTL	
74LS00N	.25
74LS02N	.25
74LS04N	.25
74LS05N	.25
74LS08N	.25
74LS10N	.25
74LS13N	.40
74LS14N	.90
74LS20N	.25
74LS22N	.25
74LS28N	.41
74LS30N	.25
74LS33N	.39
74LS38N	.30
74LS74N	1.15
74LS75N	.47
74LS90N	.51
74LS93N	.51
74LS95N	1.89
74LS107N	.35
74LS112N	.35
74LS113N	.35
74LS132N	.72
74LS136N	.35
74LS151N	.67
74LS155N	.67
74LS157N	.67
74LS162N	.91
74LS163N	.91
74LS174N	.95
74LS190N	1.06
74LS221N	1.95
74LS258N	.67
74LS367N	1.35

LINEAR	
CA3045	.90
CA3046	.67
CA3081	1.80
CA3082	1.90
CA3089	2.95
LM301AN/AH	.35
LM305H	.87
LM307N	.35
LM308N	.89
LM309H	1.15
LM309K	1.50
LM311H/N	.90
LM317T/K	2.92
LM318	1.35
LM320K-5	1.60
LM323K-5	6.95
LM320K-12	1.60
LM320K-15	1.60
LM320T-5	1.60
LM320T-8	1.60
LM320T-12	1.50
LM320T-15	1.60
LM324N	1.15
LM339N	1.55
LM340K-5	1.35
LM340K-8	1.35
LM340K-12	1.35
LM340K-15	1.35
LM340K-24	1.35
LM340T-5	1.25
LM340T-8	1.25
LM340T-12	1.25
LM340T-15	1.25
LM340T-18	1.25
LM340T-24	1.25
LM343H	4.50
LM350	7.50
LM370	1.15
LM377	3.00
LM379M	5.00
LM380N	1.00
LM381	1.60
LM382	1.60
LM703H	.40
LM709H	.28
LM723H/N	.50
LM733N	.67
LM741CH	.35
LM741N	.25
LM747H/N	.62
LM748N	.35
LM1303N	.82
LM1304	1.10
LM1305	1.27
LM1307	2.00
LM1310	2.75
LM1458	.47
LM1800	1.75
LM1812	7.50
LM1889	3.00
LM2111	1.75
LM2902	1.50
LM3900N	.60
LM3905	1.75
LM3909N	.61
MC1458V	.50
NE550N	.65
NE555V	.43
NE556A	.79
NE565A	1.00
NE566V	1.50
NE567V	1.20
NE570B	5.00
NE571B	5.00
78L05	.60
78L08	.60
79L05	.70
78M05	.85
75108	1.75
75491CN	.50
75492CN	.55
75494CN	.89

A to D CONVERTER	
8038B	4.50
8700CJ	13.95
8701CN	15.95
8750CJ	13.95
LD130	9.95
9400CJV/F	7.40
ICL7103	9.50
ICL7107	14.25
8702	17.95

CMOS	
CD34001 Fair.	.50
CD4000	.16
CD4001	.21
CD4002	.21
CD4006	1.10
CD4007	.21
CD4008	.21
CD4009	.39
CD4010	.39
CD4011	.21
CD4012	.21
CD4013	.36
CD4014	.86
CD4015	.86
CD4016	.36
CD4017	.94
CD4018	.94
CD4019	.21
CD4020	1.02
CD4021	1.02
CD4022	.86
CD4023	.21
CD4024	.75
CD4025	.21
CD4026	1.51
CD4027	.36
CD4028	.79
CD4029	1.02
CD4030	.21
CD4035	1.02
CD4040	1.02
CD4042	.71
CD4043	.63
CD4044	.63
CD4046	1.67
CD4049	.36
CD4050	.36
CD4051	1.13
CD4060	1.42
CD4066	.71
CD4068	.40
CD4069	.40
CD4070	.40
CD4071	.21
CD4072	.21
CD4073	.21
CD4075	.21
CD4076	1.75
CD4078	.40
CD4081	.21
CD4082	.21
CD4116	.47
CD4490	5.50
CD4507	1.00
CD4508	4.25
CD4510	1.02
CD4511	.94
CD4515	2.52
CD4516	1.10
CD4518	1.02
CD4520	1.02
CD4527	1.51
CD4528	.79
CD4553	3.50
CD4566	2.25
CD4583	4.50
CD4585	1.10
CD40192	3.00
74C00	.28
74C04	.33
74C10	.28
74C14	2.10
74C20	.28
74C30	.28
74C48	1.95
74C74	.75
74C76	1.40
74C90	1.15
74C93	1.40
74C154	3.00
74C160	1.44
74C175	1.35
74C192	1.65
74C221	2.00
74C905	3.00
74C906	.75
74C914	1.95
74C922	5.50
74C923	5.50
74C925	6.95
74C926	6.95
74C927	6.95

INTERFACE	
8095	.95
8096	.95
8097	.95
8098	.95
8T09	1.25
8T10	4.50
8T13	3.00
8T20	5.50
8T23	3.10
8T24	3.50
8T25	3.20
8T26	1.69
8T28	2.75
8T97	1.69
8T98	1.69

MOS/MEMORY RAM	
2101-1	3.95
2102-1	.95
2102AL-4	1.60
21L02-1	1.18
21F02	1.25
2104A-4	4.95
2107B	4.95
2111-1	3.75
2112-2	3.95
2114L-3	6.75
4116	10.95
2513B	6.30
MM5262	.40
MM5280	3.00
MM5320	9.95
MM5330	5.94
PD411D-3	4.00
PD411D-4	5.00
P5101L	13.95
4200A	9.95
82S25	2.90
91L02A	1.50
HD0165-5	6.95
MM57100	4.50
GIAY38500-1	9.95
MCM6571A	9.95
9368	3.50
410D	10.00
416	16.00

CLOCKS	
MM5309	3.00
MM5311	3.60
MM5312	4.80
MM5313	3.60
MM5314	3.90
MM5315	4.00
MM5316	5.00
MM5318	3.60
MM5369	2.10
MM5841	14.45
MM5865	7.95
CT7001	5.80
CT7002	10.95
CT7010	8.95
CT7015	8.95
MM5375AA/N	3.90
MM5375AB/N	4.90
7205	16.50
7207	7.50
7208	15.95
7209	4.95
DS0026CN	3.75
DS0056CN	3.75
MM53104	2.50

MICROPROCESSOR	
6800	17.50
6802	24.95
8080A with data	8.95
8085	27.00
Z80A	19.75
8212	2.90
8214	8.00
8216	2.90
8224	2.90
8228	5.35
8251	8.50
8253	10.00
8255	9.25
8257	19.50
8259	19.50
1802CP plas.	13.95
1802DP plas.	17.95
1861P	11.50
CDP1802CD	19.95
CDP1802D	25.00
CDP1861	12.95
6820	9.95
6850	12.95
6502	12.50

IC SOCKETS
Solder Tin Low Profile

PIN	1 UP	PIN	1UP
8	.15	22	.30
14	.14	24	.35
16	.16	28	.42
18	.27	36	.58
20	.29	40	.57

3 level wire wrap gold. 14 pin .25
2 level 14 pin ww .20

WIRE WRAP LEVEL 3

PIN		PIN	
14	.25	24	.86
16	.33	28	1.00
18	.57	40	1.23

UART/FIFO	
AY5-1013	5.50
AY5-1014	7.50
3341	6.95

PROM	
1702A	3.95
N82S23	2.95
N82S123	3.50
N82S126	3.75
N82S129	3.75
N82S131	3.75
N82S136	8.75
N82S137	8.75
2708	10.50
DM8577	2.90
8223	2.90
2716T1	29.50
2716 Intel	48.00

CRYSTALS	
1 MHz	4.50
2 MHz	4.50
4 MHz	4.25
5 MHz	4.25
10 MHz	4.25
18 MHz	3.90
20 MHz	3.90
32 MHz	3.90
32768 MHz	4.00
1.8432 MHz	4.50
3.5795 MHz	1.20
2.0100 MHz	1.95
2.097152 MHz	4.50
2.4576 MHz	4.50
3.2768 MHz	4.50
5.0688 MHz	4.50
5.185 MHz	4.50
5.7143 MHz	4.50
6.5536 MHz	4.50
14.31818 MHz	4.25
18.432 MHz	4.50
22.1184 MHz	4.50

CONNECTORS	
44 pin edge	2.75
100 pin edge	4.50
100 pin edge WW	4.75

KEYBOARD ENCODERS	
AY5-2376	$12.50
AY5-3600	13.50
74C922	5.50
74C923	5.50
HD0165-5	6.95

IC Test Clips	1	10
Red	.55	.47
Black	.55	.47

Keyer 8043 14.50 comp. w/spec./sock.

TRANSISTORS	
2N1893	.40
2N2222A	.18
2N2369	.30
2N2904A	.20
2N2907A	.25
2N3053	.40
2N3638	.25
2N3643	.25
2N3904	.18
2N3906	.18
2N3055	.69
2N4400	.25
2N4401	.75
2N4402	.20
TIP31	.60
TIP33A	1.00

D Connectors RS232
25 Pin Subminiatures

DB25P	2.95
DB25S	3.95
Cover	1.50
RS232 Complete Set	6.50
DE9P	1.50
DE9S	1.95
DA15P	2.10
DA15S	3.10

CLOCK MODULES Complete alarm clocks ready to hook up with transformer and switches. Very compact with .50" and .84" digits.

MA1002A, C or E .50"	**8.95**
102P3 Transformer	**2.25**
MA1010A, C or E .84"	**11.95**
102P2 Transformer	**2.25**
Special transformer and six switches when purchased w/module	**2.95**
MA1003 car module .3" green fluor. display	**15.95**

RESISTORS ¼ watt 5%
10 per type .03 1000 per type .012
25 per type .025 350 piece pack
100 per type .015 5 per type 6.75

KEYBOARDS	
56 key ASCII keyboard kit	$67.50
Fully assembled	77.50
53 key ASCII keyboard kit	60.00
Fully assembled 70.00 Enclosure	14.95

LEDS	
Red T018	.15
Green, Yellow T018	.20
Jumbo Red	.20
Green, Orange, Yellow Jumbo	.25
Cliplite LED Mounting Clips	8/$1.25

(specify red, amber, green, yellow, clear)

CONTINENTAL SPECIALTIES in stock
Complete line of breadboard test equip.
MAX-100 8 digit Freq. Ctr. $128.95

OK WIRE WRAP TOOLS in stock
Portable Multimeter $18.00

DIGITAL THERMOMETER $48.50
Batt. oper. General purpose or medical 32 – 230°F. Disposable probe cover ±.2 accuracy. Comp. Assy. in compact case. Switches from F° to C°.

COMPUTER BOARD KITS	
8K RAM Board Kit	$135.00
4K EPROM Kit	114.95
I/O Board Kit	44.50
Extender Board w/connector	12.50
16K EPROM board kit w/o PROMS	74.50
North Star Floppy Disk Kit	$665.00
Additional Drive Kit	415.00

SPECIAL PRODUCTS	
MM5865 Stopwatch Timer	9.00
PC board	7.50
Switches Mom. Pushbutton	.27
3 pos. slide	.25
Encoder HD0165-5	6.95
3 Digit Universal Counter Board Kit Operates 5–18 Volt DC to 5 MHz typ. .125" LED display	**10.50**
Voice actuated switch	.50
Paratronics 100A Logic Analyzer Kit	$235.00
Model 10 Trigger Expander Kit	$229.00
Model 150 Bus Grabber Kit	$369.00
Sinclair 3½ Digit Multimeter	**$59.95**
Clock Calendar Kit	**$23.95**

TRANSFORMERS	
6V 300 ma	3.25
12 Volt 300 ma transformer	1.25
12.6V CT 600 ma	3.75
12V 250 ma wall plug	2.95
12V CT 250 ma wall plug	3.50
24V CT 400 ma	3.95
10V 1.2 amp wall plug	4.85
12V 6 amp	12.95

DISPLAY LEDS			
MAN1	CA	.270	2.90
MAN3	CC	.125	.39
MAN72/74	CA/CA	.300	1.00
DL704	CC	.300	1.25
DL707/DL707R	CA	.300	1.00
DL727/728	CA/CC	.500	1.90
DL747/750	CA/CC	.600	1.95
DL750	CC	.600	1.95
FND359	CC	.357	.70
FND500/507	CC/CA	.500	1.35
FND503/510	CC/CA	.500	.90
FND800/807	CC/CA	.800	2.20
3 digit Bubble			.60
4 digit bubble			.80
DG8 Fluorescent			1.75
DG10 Fluorescent			1.75
5 digit 14 pin display			1.00
NSN69 9 digit display			.60
7520 Clairex photocells			.39
TIL311 Hex			9.50

COMPUTER GRADE CAPS			
1600	mfd	200V	4.50
2000	mfd	45V	2.50
3200		50V	2.50
5500		25V	2.50
5800		40V	3.00
6100		40V	3.00
6100		50V	3.00
7700		50V	2.50
8000		55V	2.50
10000		20V	2.50
12000		30V	3.50
35000		25V	5.00
55000		25V	5.00
82000		15V	6.00
60000		40V	6.00
76000		40V	6.00
120000		10V	6.00

RCA Cosmac Super Elf Computer $106.95

Compare features before you decide to buy any other computer. There is no other computer on the market today that has all the desirable benefits of the **Super Elf** for so little money. The Super Elf is a small single board computer that does many **big** things. It is an excellent computer for training and for learning programming with its machine language and yet it is easily **expanded** with **additional memory, Tiny Basic, ASCII Keyboards, video character generation, etc.**

The **Super Elf** includes a **ROM** monitor for program loading, editing and execution with **SINGLE STEP for program debugging** which is not included in others at the same price. With **SINGLE STEP** you can see the microprocessor chip operating with the **unique Quest** address and data bus displays **before, during** and **after** executing instructions. Also, CPU mode and instruction cycle are decoded and displayed on eight LED indicator lamps.

An RCA 1861 **video graphics** chip allows you to connect to your own TV with an inexpensive video modulator to do graphics and games. There is a **speaker system** included for writing your own music or using many music programs already written. The speaker amplifier may also be used to drive relays for control purposes.

A 24 key **HEX keyboard** includes 16 HEX keys plus **load, reset, run, wait, input, memory protect, monitor select** and **single step**. Large, on board displays provide output and optional **high and low address**. There is a 44 pin standard connector for PC cards and a 50 pin connector for the **Quest Super Expansion Board**. Power supply and sockets for **all** IC's are included in the price plus a detailed 127 pg. instruction manual which now includes over 40 pgs. of software info. including a series of lessons to help get you started and a music program and graphics target game.

Remember, other computers only offer Super Elf features at additional cost or not at all. **Compare before you buy. Super Elf Kit $106.95, High address option $8.95, Low address option $9.95. Custom Cabinet** with drilled and labelled plexiglass front panel **$24.95. NiCad Battery Memory Saver Kit $6.95.** All kits and options also come completely assembled and tested.

Questdata, a 12 page monthly software publication for 1802 computer users is available by subscription for $12.00 per year.

Attention Elf Owners

New products in hardware and software coming soon.

Tiny Basic cassette $10.00, on ROM $38.00, original Elf kit board $14.95.

Super Expansion Board with Cassette Interface $89.95

This is truly an astounding value! This board has been designed to allow **you** to decide how you want it optioned. **The Super Expansion Board comes with 4K of low power RAM** fully addressable anywhere in 64K with built-in memory protect and a **cassette interface.** Provisions have been made for all other options on the same board and it fits neatly into the hardwood cabinet alongside the **Super Elf.** The board includes slots for up to 6K of **EPROM** (2708, 2758, 2716 or TI 2716) and is **fully socketed.** EPROM can be used for the monitor and Tiny Basic or other purposes.

A 1K Super ROM Monitor $19.95 is available as an on board option in 2708 EPROM which has been preprogrammed with a program loader/editor and error checking multi file cassette read/write software, (relocatible cassette file) another exclusive from Quest. It includes register save and readout, block move capability and video graphics driver with blinking cursor. Break points can be used with the register save feature to isolate program bugs quickly, then follow with single step. The Super Monitor is written with subroutines allowing users to take advantage of monitor functions simply by calling them up.

Improvements and revisions are easily done with the monitor. If you have the **Super Expansion Board** and **Super Monitor** the monitor is up and running at the push of a button.

Other on board options include **Parallel Input and Output Ports** with **full handshake.** They allow easy connection of an ASCII keyboard to the input port. **RS 232 and 20 ma Current Loop** for teletype or other device are on board and if you need more memory there are two **S-100** slots for static RAM or video boards. A Godbout 8K RAM board is available for $135.00. Also a 1K Super Monitor version 2 with video driver for full capability display with Tiny Basic and a video interface board. **Parallel I/O Ports $9.85, RS 232 $4.50, TTY 20 ma I/F $1.95, S-100 $4.50. A 50 pin connector set** with ribbon cable is available at $12.50 for easy connection between the **Super Elf** and the **Super Expansion Board.**

The **Power Supply Kit** for the Super Expansion Board is a 5 amp supply with multiple positive and negative voltages $29.95. Add $4.00 for shipping. Prepunched frame $5.00. Case $10.00. Add $1.50 for shipping.

Auto Clock Kit $15.95

DC clock with 4-.50" displays. Uses National MA-1012 module with alarm option. Includes light dimmer, crystal timebase PC boards. Fully regulated, comp. instructs. Add $3.95 for beautiful dark gray case. Best value anywhere.

RCA Cosmac VIP Kit $229.00

Video computer with games and graphics. Fully assem. and test. **$249.00**

Not a Cheap Clock Kit $14.95

Includes everything except case. 2-PC boards. 6-.50" LED Displays. 5314 clock chip, transformer, all components and full instructions. Orange displays also avail. **Same kit w/.80" displays.** Red only. **$21.95** Case **$11.75**

60 Hz Crystal Time Base Kit $4.40

Converts digital clocks from AC line frequency to crystal time base. Outstanding accuracy. Kit includes: PC board, IC, crystal, resistors, capacitors and trimmer.

Digital Temperature Meter Kit

Indoor and outdoor. Switches back and forth. Beautiful. 50" LED readouts. Nothing like it available. Needs no additional parts for complete, full instructions. Will measure −100° to +200°F, tenths of a degree, air or liquid. Very accurate. **$39.95**
Beautiful woodgrain case w/bezel **$11.75**

NiCad Battery Fixer/Charger Kit

Opens shorted cells that won't hold a charge and then charges them up, all in one kit w/full parts and instructions. **$7.25**

PROM Eraser

Will erase 25 PROMs in 15 minutes. Ultraviolet, assembled **$34.50**

Rockwell AIM 65 Computer

6502 based single board with full ASCII keyboard and 20 column thermal printer. 20 char. alphanumeric display, ROM monitor, fully expandable. $375.00. 4K version $450.00. 4K Assembler $85.00, 8K Basic Interpreter $100.00. Power supply assembled in case $60.00.

Multi-volt Computer Power Supply

8v 5 amp, ±18v .5 amp, 5v 1.5 amp, −5v .5 amp, 12v .5 amp, −12 option. ±5v, ±12v are regulated. Kit $29.95. Kit with punched frame $34.95. Woodgrain case $10.00.

Video Modulator Kit $8.95

Convert your TV set into a high quality monitor without affecting normal usage. Complete kit with full instructions.

2.5 MHz Frequency Counter Kit

Complete kit less case **$37.50**

30 MHz Frequency Counter Kit

Complete kit less case **$47.75**
Prescaler kit to 350 MHz **$19.95**

79 IC Update Master Manual $35.00

Complete IC data selector, 2500 pg. master reference guide. Over 50,000 cross references. Free update service through 1979. Domestic postage $3.50. Foreign $5.00. 1978 IC Master closeout $19.50.

Stopwatch Kit $26.95

Full six digit battery operated. 2–5 volts. 3.2768 MHz crystal accuracy. Times to 59 min., 59 sec., 99 1/100 sec. Times std., split and Taylor. 7205 chip, all components minus case. Full instructions.

Hickok 3½ Digit LCD Multimeter

Batt/AC oper. 0.1mv–1000v. 5 ranges. 0.5% accur. Resistance 6 low power ranges 0.1 ohm–20M ohm. DC curr. .01 to 100ma. Hand held, ½" LCD displays, auto zero, polarity, overrange. $74.95.

S-100 Computer Boards

8K Static RAM Kit Godbout	$135.00
16K Static RAM Kit	265.00
24K Static RAM Kit	423.00
32K Dynamic RAM Kit	310.00
64K Dynamic RAM Kit	470.00
8K/16K Eprom Kit (less PROMS)	**$89.00**
Video Interface Kit	**$139.00**
Motherboard **$39.** Extender Board	**$8.99**

TERMS: $5.00 min. order U.S. Funds. Calif residents add 6% tax.
BankAmericard and Master Charge accepted.
Shipping charges will be added on charge cards.

FREE: Send for your copy of our NEW 1979 QUEST CATALOG. Include 28¢ stamp.

Circle 311 on inquiry card.

Circle 387 on inquiry card.

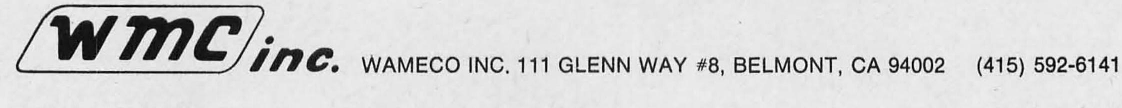
WAMECO

THE COMPLETE PC BOARD HOUSE
EVERYTHING FOR THE S-100 BUSS

* **FPB-1** FRONT PANEL BOARD
  Hex Displays, IMSAI Replaceable $54.95
* **FDC-1** FLOPPY DISC CONTROLLER BOARD
  Controls up to 8 Discs $45.00
* **MEM-1A** 8K BYTE 2102 RAM Board $31.95
* **MEM-2** 16K BYTE 2114 RAM Board $31.95
* **CPU-1** 8080A CPU Board
  With Vector Interrupt $31.95
* **EPM-1** 4K BYTE 1702A EPROM $29.95
* **EPM-2** 16K or 32K BYTE EPROM
  2708 or 2176 interchangeable $30.00
* **QMB-9** 9 SLOT MOTHER BOARD
  Terminated $35.00
* **QMB-12** 12 SLOT MOTHER BOARD
  Terminated $40.00
* **RTC** REALTIME CLOCK
  Programmable Interrupts $27.95

FUTURE PRODUCTS: 80 CHARACTER VIDEO BOARD,
IO BOARD WITH CASSETTE INTERFACE.

**DEALER INQUIRIES INVITED, UNIVERSITY DISCOUNTS AVAILABLE
AT YOUR LOCAL DEALER**

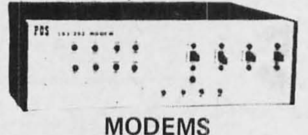


COMMERCIAL GRADE PERIPHERALS FOR THE MICROCOMPUTER

PRINTER TERMINALS









TAPE DRIVES

MODEMS

•**ASCII SELECTRIC PRINTER/TYPEWRITER:** Why settle for less than letter-quality printout from your computer? Refurbished IBM Model 725 can be used as off-line typewriter or on-line printer. Complete with solenoids, power supply, case and ASCII interface card (TTL to CPU parallel port.) Interface includes programmable ASCII translation table on EPROM with up to 8 tables for use with various type spheres. Feedback signals on completion of each print cycle insures fastest printing speed (15 cps.)
Price: programmed w/3 translation tables (one type sphere): $695.00

•**SELECTRIC I/O TERMINALS** (by GTE/Information Systems). Both ASCII & IBM code versions with microcomputer interface software & hardware (RS-232 connector.) Cassette drive models permit up to 2400 baud data transfer rate as well as off-line data storage, use as memory typewriter, & use as data entry device for office personnel familiar with Selectric typewriters but not computers. Wide-carriage, interchangeable type spheres; optional built-in modem. All units cleaned, adjusted & warranted.
Model 5541 (IBM Correspondence code) $695.00
Model 5550 (corres. code, built-in cassette drive) $1195.00
Model 5560 (ASCII code, built-in cassette drive) $1295.00

•**IBM SELECTRIC 725 TYPEWRITER I/O** w/solenoids switches & magnet driver PCB (from GTE/IS terminal) plus instructions for 8080 printer-driver interface hardware & software.
a) Typewriter mechanism complete, cleaned & adjusted $375.00
b) Case from terminal & power supply (+24V, ±12V, +5V @5A). . $ 75.00

•**DIABLO HYTYPE I Model 1200 PRINTER MECHANISM:** used, complete and tested. Requires power supply, case & mCPU interface. 15 day return privilege – no other warranties. LIMITED QUANTITY! $750.00
–6' Ribbon cable & connector for printer Main Logic PCB $10.00
–14-pin Winchester connector & 18" power supply cable $5.00
–"As-is" spare printer PCB's for parts (Logic, Heat Sink, Control): ea. $20.00
–New Pin-feed Platen (14"): $50 if bought w/printer; separately ... $100.00

•**POS 103/202** *"MIX or MATCH"* **MODEM:** BELL 103 and/or BELL 202 FREQUENCIES: Unique POS control design permits use in one housing of both Bell-compatible 103 (0 – 300 baud) and 202 (0 – 1200 baud) modem modules originally made by VADIC Corp. for a telephone company subsidiary. FEATURES: RS-232 serial interface, auto-answer, auto-dial, LED display, telephone line interface via acoustic coupler, manual DAA, or auto-answer DAA (sold separately.) FULLY ADJUSTED; no special tools required. 3,000 mile range over standard dial-up telephone lines.
–POS 103 MODEM (with Auto Answer, Auto Dial) $199.95
–POS 202 MODEM (Half-Duplex with Reverse Channel). $249.95
–POS 202 MODEM (Half-Duplex w/Rev. Ch., Auto-Answer)...... $279.95
–POS 103/202 MODEM (Auto-Answer, Auto-Dial). $399.95

•**POS-100 NRZ1 TAPE DRIVE CONTROLLER/FORMATTER:** Designed as interface between S-100 bus mCPU and 9-track, 800 BPI, NRZ1 tape drive. Allows microcomputerist to read and write IBM-compatible ½" mag tapes. Software provided for 8080 or Z-80 systems. Requires modification for drives of various mfrs.
Price: (Includes S-100 card, controller card, 10' cable, software listing) . $750.00

•**NRZ1 TAPE DRIVE by WILLARD LABS.** 9-track, 800 BPI, NRZ1 format, 12"/sec., 1200 ft. reels (10 megabyte capacity) Fully tested and warranted $599.00

•**CONVERT 15" IBM OFFICE SELECTRIC TO I/O TYPEWRITER:** Kit includes assembled solenoids, switches, wire harness, magnet driver PCB plus instructions for installation and mCPU interface $200.00

•**DIGITAL CASSETTE DRIVE** (from GTE/IS Terminal): 1800 baud, 6"/sec; AC motor; fwd/rewnd circuitry plus tape head, no read/write electronics $25.

•**FORMS TRACTORS,** Moore Variable width "Form A-Liner" for print terminals:
a) Model 565P for 15" Carriage IBM Selectrics (new): $50.00
b) Model K81 for QUME or DIABLO Hytype I or II printers (new): $90.00

•**POWER SUPPLIES** for Disk Drive, mCPU, tested under load shown:
–No. 519 (w/fan & AC cord): +5V reg., ±12V reg., +24V, @4A (10 lb.). . $29.95
–LAMBDA No. LMEE5 w/OV protect: +5V reg. @ 25A (35 lb.). $59.95

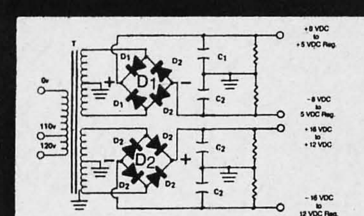
NO RISK! 15 DAY APPROVAL ON ALL MAIL-ORDERS

Full documentation included PLUS interface instructions where indicated. All equipment is shipped insured FOB Palo Alto within 14 days after check clears or COD order is received. Prices may change without notice.

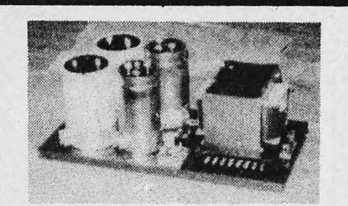
PACIFIC OFFICE SYSTEMS, INC.
2600 El Camino Real, Suite 502
Palo Alto, Calif. 94306
(415) 321-3866

Call or write for details, quantity prices, catalog. 15 day return privilege PLUS 90 day no charge replacement of defective parts. All orders shipped from stock. No back orders, no substitutions. M/C & VISA accepted.

Circle 296 on inquiry card.



BUILD YOUR OWN LOW COST MICRO-COMPUTER POWER SUPPLIES FOR S-100 BUS, FLOPPY DISCS, ETC.



POWER TRANSFORMERS (WITH MOUNTING BRACKETS)

ITEM NO.	USED IN KIT NO.	PRI. WINDING TAPS	SECONDARY WINDING OUTPUTS 2×8 Vac	2×14 Vac	2×24 Vac	SIZE W × D × H	UNIT PRICE
T_1	1	0V, 110V, 120V	2×9A	2×2.5A	———	3¾"×3⅝"×3⅛"	19.95
T_2	2	0V, 110V, 120V	2×12.5A	2×3.5A	———	3¾"×4⅜"×3⅛"	25.95
T_3	3	0V, 110V, 120V	2×10A	2×2.5A	2×2.5A	3¾"×4⅜"×3⅛"	27.95
T_4	4	0V, 110V, 120V	2×4.5A	———	2×4.5A	3¾"×3⅝"×3⅛"	19.95

POWER SUPPLY KITS (OPEN FRAME WITH BASE PLATE, 3 HRS. ASSY. TIME)

ITEM	USED FOR	@+8 Vdc	@−8 Vdc	@+16 Vdc	@−16 Vdc	@+28 Vdc	SIZE W×D×H	UNIT PRICE
KIT 1	18 CARDS SOURCE	18A	———	2.5A	2.5A	———	12"×6"×4⅞"	46.95
KIT 2	SYSTEM SOURCE	25A	———	3A	3A	———	12"×6"×4⅞"	54.95
KIT 3	DISC SYSTEM	18A	1A	2A	2A	4A	14"×6"×4⅞"	62.95
KIT 4	DISC SOURCE	8A	1A	———	———	8A	10"×6"×4⅞"	44.95

EACH KIT INCLUDES: TRANSFORMER, CAPACITORS, RESIS., BRIDGE RECTIFIERS, FUSE & HOLDER, TERMINAL BLOCK, BASE PLATE, MOUNTING PARTS AND INSTRUCTIONS.

REGULATED POWER SUPPLY "R1" ASSY. & TESTED, OPEN FRAME, SIZE: 9" (W) x 5" (D) x 5" (H) **$49.95**

SPECS: +5V, ±1%, @ 5A, OVERCURRENT PROTECTION, 5% ADJ. FOR +5V. +24V UNREG. @ 5A FOR PRINTERS OR DISCS.

REMARK: IDEAL FOR ROCKWELL AIM-65 MICROCOMPUTER. ALSO UNREG. +8V @ 8A ATTAINABLE FOR USE IN S-100 BUS.

SHIPPING FOR EACH TRANSFORMER: $4.75. FOR EACH POWER SUPPLY: $5.00 IN CALIF. $7.00 IN OTHER STATES. CALIF. RESIDENTS ADD 6% SALES TAX. OEM WELCOME.

VISA

MAIL ORDER:
P.O. BOX 4296
TORRANCE, CA 90510

SUNNY INTERNATIONAL
(TRANSFORMERS MANUFACTURER)
Telephone: (213) 633-8327

STORE:
7245 E. ALONDRA BLVD.
PARAMOUNT, CA 90723
STORE HOURS: 9 AM–6 PM



COMPUCOLOR II IS HERE!!

HERE IS A SURPRISINGLY AFFORDABLE COMPUTER THAT MAKES THE COMPETITION LOOK TWICE!!

STANDARD FEATURES

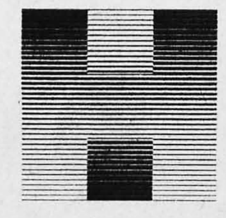
- 13" COLOR CRT
- SPECIAL GRAPHICS PKG.
- EXTENDED DISK BASIC
- MINI DISK DRIVE
- 8K RAM MEMORY
- 72 KEY KEYBOARD

PRICE & OPTIONS

MODEL 3 — 8K USER RAM - 1495.00
MODEL 4 — 16K USER RAM - 1695.00
MODEL 5 — 32K USER RAM - 1995.00
2nd DISK DRIVE - 400.00
EXPANDED KEYBOARD - 135.00
FORMATTED DISKETTE - 5.00
MANY PROGRAMS AVAILABLE *

TO ORDER

SEND CHECK OR MONEY ORDER
CALIF. RES. add 6% TAX
shipping 1% all orders
CATALOG — 50¢



HOLLYWOOD SYSTEMS

9100 SUNSET BLVD.
SUITE 112
L.A. CALIF. 90069

Unclassified Ads

NEW UNCLASSIFIED POLICY

Readers who have equipment, software or other items to buy, sell or swap should send in a clearly typed notice to that effect. To be considered for publication, an advertisement must be clearly noncommercial, typed double spaced on plain white paper, contain 75 words or less, and include complete name and address information.

These notices are free of charge and will be printed one time only on a space available basis. Notices can be accepted from individuals or bona fide computer users clubs only. We can engage in no correspondence on these and your confirmation of placement is appearance in an issue of BYTE.

Please note that it may take three or four months for an ad to appear in the magazine.■

MUST SELL: SwTPC 6800 computer system, includes 16 K programmable memory, two serial ports, PerCom Data CIS-30 plus cassette storage unit, software and manuals. $525 buys everything; shipped post paid. Don Domek, 535 Burlington, Billings MT 59101, (406) 248-3477.

FOR SALE: One SD sales expandoram set up with integrated circuits for 16 K (for 115-41); has problem. Data can be entered and retrieved but programs will not run, $200. CFI add-on memory for IBM system III, mod 6 or 10, 16 K. Working condition, with cables, $2000 or best offer. Norm Doty, 53 Kaufman Rd, Cheektowaga NY 14225, (716) 892-8829.

FOR SALE: KIM system KIM-1 Computer, Enclosures Group case, miniature 4 K wire wrap memory (21LO2s), TVT-6 video interface, surplus video monitor, Radio Shack ASCII keyboard, oscillator/driver board and speaker, I/O (input/output) connector block, 5 V, 1 A supply, 5 V, 2 A +/12 V, 1 A supply. All for $450. Ron Kushnier, 3108 Addison Ct, Cornwells Heights PA 19020, (215) 757-9057.

FOR SALE: IMSAI 4 K static, $80. MITS 4 K dynamic, $50. Tarbell cassette board, $90. MITS ACR, $70. MITS disk drive with BASIC and FORTRAN, $1,300. MITS-Okidata printer with controller, $1,500. All boards fully socketed and factory checked out. Make offer. For trade: 8080 FORTRAN MITS disk version for 8080 COBOL or MITS timesharing BASIC. Manuals available. K R Roberts, 10560 Main St, Suite 515, Fairfax VA, 22030, (703) 591-6008 or 378-7266.

FOR SALE: SwTPC MF-68 minifloppy disk system, $850. CT-64 terminal with two pages memory, screen read board, etc. Also CT-VM monitor, $450. Two MP-8M memory boards, $190 each. Expandor black box printer with base, cover and case of paper, $390. Will make good price on TSC disk software to purchaser of minifloppy. All units are complete with documentation, were assembled by a professional and are 100 percent functional. John Gorman, 143 Chenault Rd, Lexington KY 40502.

FOR SALE: Digital Group Z-80 26 K, dress cabinet, 64 character TV controller, two Phi-Decks and controller, PHIMON, Business BASIC etc. Original cost over $3,000. Will sell for $1,995. William C Dewberry Jr, 314 Interbay Av, Pensacola FL 32597, (904) 456-1061.

FOR SALE: A complete computer system consisting of an Altair 8800-A with 36 K of static programmable memory, a North Star 5 inch floppy disk drive, VDM-1, 3P+S, Cherry Switch keyboard, ALS-8, Sanyo video monitor, Bytesaver, and a real time clock. Highest offer above $1700 accepted. Also, Sol motherboard computer for $500. Juan Rivera, 354 Marshall Dr, Walnut Creek CA 94598, (415) 935-3235.

FOR SALE: Used Selectric I/O (input/output) printer Model 731 in working condition, $450. TVT II board assembled with full data $50. Core memory plane 4 K by 16 bits or 8 K by 8 with interface and drive information $50. Shipping extra. Ted Becker, 317 158th St SE, Bothell WA 98011, (206) 743-1321.

WANTED: Back issues of BYTE, Interface Age, Popular Electronics, Popular Mechanics, Radio Electronics and Scientific American in good condition. Send details of holdings and price. Also, I have some duplicates of the above magazines to sell or swap. SASE would be appreciated. Michael Carter, 62B Escondido Village, Stanford CA 94305.

FOR SALE: AMD9511 arithmetic integrated circuit on S-100 card with BASIC-E. Calculate SIN (X) in 2.8 ms in BASIC. $250. G Lyons, 280 Henderson St, Jersey City NJ 07302, (201) 451-2905.

WANTED: Microcomputers; TRS-80, PET, Apple; any condition and quantity. Immediate cash available or trade for DEC PDP8e, f, m. 4 K PDP8m with teletypewriter interface $1100. Portacom briefcase ASCII terminals with modems $595. K2DCY. Box 632, W Caldwell NJ 07006, (201) 226-9185.

FOR SALE: Digital Group 10 K Z-80 system, assembled, includes MINI-BASIC, MAXI-BASIC, Assembler II, Editor, Op-sys, and all documentation. Asking $1000. Will consider selling components separately. Also 8 K of 21L02 integrated circuits, 450ns; unused, $65. J E Tucker, Box 4338, APO NY 09223.

FOR SALE: Surplus pipe organ parts. Direct electric chests, keyboards, console, cable, and various ranks of pipes. Low pressure, good to excellent condition with reasonable to ridiculously low prices. Also Friden Flexowriter, Potter high speed photo tape reader, 8 level punch and reader with power supply and interface electronics. Ferranti tape spoolers, other readers. Prefer pickups. Andy LaTorre, 45 Ellis Av, Northport NY 11768, (516) 757-1913.

FOR SALE: Vista 5 1/4 inch floppy disk drive with S-100 controller and cables. Factory assembled, mint condition. Too small for my needs. Includes CP-M, BASIC-E compiler, VOS assembler, text editor, other utilities, plus five diskettes of software and games. Paid $750, asking $500. Also, new cabinet and power supply for above. Paid $80, asking $50. David Schwinck, 2221 SW 14th, Lincoln NE 68522.

WANT TO TRADE: Have a Kenwood KT5300 tuner and KA7300 integrated amplifier with CT-F9191 Pioneer cassette and Infinity 2000 II speakers in excellent condition. Worth $2600 new. Would like a good computer system with mini disk of similar value. Leroy A McDaniel, 916 N 4th St, McAllen TX 78501.

FOR SALE: Complete assembled and running microcomputer system. Includes 680b mother board with cabinet and power supply, 16 K static memory, 680 KCACR cassette interface, 8 K BASIC on cassette, Assembler and Editor on paper tape, SwTPC. CT-1024 terminal system complete with all options plus cabinet for keyboard, scrolling conversion kit, and fully socketed with 74LS series integrated circuits. Asking $1000 US. R Pieracci, 43 Mayfield Rd, Regina Saskatchewan CANADA, S4V 0B7.

WANTED: Manuals for Wang BAS system. Also, software suitable for Wang 2200C cassette based computer. J E Thompson, POB 128, Monee IL 60449.

FOR SALE: Four 8 K memory boards for Heathkit H8(WH8-8); assembled, tested and burned in. Heathkit price, $250 each; asking $195 each or four for $750. Henry Fale, 2918 S 7th St, Sheboygan WI 53081, (414) 452-4172.

FOR SALE: Intel SDK-85 development kit; fully assembled. 3 K programmable memory, cassette interface (software in 2708). Powermate PS 5 V/6 A. All integrated circuits socketed; contain all connectors. Packaged inside attache case. Full documentation. First money order for $375 gets it; I pay shipping. L Stroll, 211-05 85th Av, Hollis Hills NY 11427, (212) 464-7341.

FOR SALE: SwTPC M6800 with 8 K memory $300; MP-A processor board $100; AC-30 $65; 4 K memory boards $60; MSI 2708 erasable read only memory board $85; Digital Group Phi-Deck system, two drives, controller (not working), software, $250: above items fully socketed. Phillips Digital cassette drives, simple interface $125; Heathkit oscilloscope I/O (input/output) 105 15 MHz, dual trace $425; Sola constant voltage xfmer, 750 W $150. Jim Georgoulis, 504 Fort Drum Dr, Austin TX 78745, (512) 441-6568.

FOR SALE: Two MITS Altair 4 K memory boards, model 88-4MCD, assembled and in service now, no bad bits. Includes original documentation. I need the slots. Price $75 each, postpaid and insured. Money order or certified check, or allow three weeks for personal check to clear. Lewis Mosley Jr, 2576 Glendale Ct NE, Conyers GA 30208.

FOR SALE: IMSAI microcomputer with 28 K programmable memory, read only memory board, Tarbell cassette interface, poly video board, keyboard, monitor, and cassette tape recorder. All documented and working, $1300. Leo Breiman, 905 Centinela Av, Santa Monica CA 90403, (213) 828-2840 or 829-7411.

FOR SALE: Seattle Computer Products 16 K plus static memory, $350; Tarbell Electronics floppy disk controller, $225. Both new, factory assembled and tested units. Glenn Nelson, 205 Meadows Rd, Whitefish MT 59937, (406) 862-3854.

FOR SALE OR TRADE: S D Sales, 16 K Expandoram (has sockets for 32 K). This board will not work with DMA or any application that requires wait states. I prefer to trade for static memory. I will sell for $200 ONO. Wayne Miller, 905 Fairmount Blvd, Jefferson City MO 65101.

FOR SALE: One Processor Technology VDM-1 board kit, $130; one Processor Technology 8 K programmable memory board kit, $170. Both kits in original factory package — never opened. Best offer. Joe Haran, 607 Painters Xing, Chadds Ford PA 19317, (215) 358-3346.

FOR SALE: Assembled Cromemco D+7A analog I/O (input/output) board. Ribbon cable and edge connector included. I will include all driving and applications software I have written to date. Excellent condition, only $112. John Peterson, 1820 Camino Dr, Forest Grove OR 97116.

FOR SALE: 8 K programmable memory board, S-100, fully buffered. Memory protected, NEC2102L-450. Used over 50 hours; $129. J Grina, 1284 Fifield Pl, St Paul MN 55108.

FOR SALE: Model 123P Expandor/Mite printer. Complete with case and interface electronics. Ready to plug into any 8 bit parallel port. Full documentation. $325 postpaid. A P Stumpf, Box 1603, Litchfield Park AZ 85340, (602) 935-2053.

FOR SALE: A limited number of BYTE, Volume 1 #1, in mint condition. Never been opened. Highest bids can have them while they last. Joe Haran, 607 Painters Xing, Chadds Ford PA 19317, (215) 358-3346.

FOR SALE: Digital Group Z-80 26 K system, keyboard, dual Phi-Decks, printer. All working, in dress cabinets; lots of software. No reasonable offer refused. Scott Bishop, 2221 Charlotte Dr, Maitland FL 32751, (305) 869-4203.

Reader Service

To get further information on the products advertised in BYTE, fill out the reader service card with your name and address. Then circle the appropriate numbers for the advertisers you select from the list. Add a 15 cent stamp to the card, then drop it in the mail. Not only do you gain information, but our advertisers are encouraged to use the marketplace provided by BYTE. This helps us bring you a bigger BYTE.

Inquiry No.	Page No.
1	Aaron Associates 205
2	Administrative Systems 150
4	Advanced Computer Products 251
5	American Square Computers 245
7	Anadex 82,83
15	Apple Computer 19
10	apple tv & computing 247
11	ASAP Computer Products Inc 283
19	ATV Research 277
20	Avery Labels 231
21	Avionic Enterprises Inc 245
22	base 2, inc. 101
23	Basic Time 175
30	Beckian Enterprises 282
27	Beta Business Systems 277
*	Bits Inc 181, 200, 211
33	Body Peripherals 122
34	Broadcast Computer Systems 231
31	Business Applications Software 163
32	Buss/Charles Floto 215
*	BYTE Back Issues 217
36	BYTE Books 113, 125-128, 133, 199, 248
37	California Computer Systems 22, 23
39	California Digital 259
40	Cambridge Development Labs 222
45	Central Data 111
46	Chatsworth Data 12
47	Chrislin Industries 207
50	"Compucolor" (Div Intelligent Sys) 27
51	Compucolor (Div Intelligent Sys) 69
53	CompuMax Associates 141
52	The Computer Cookbook 63
*	Computer Factory NY 276
*	Computer Headwear 227
*	Computer Lab NJ 239
75	Computerland 8, 9, 274
66	Computer Mart of NJ & PA 21
67	Computer Packages Unlimited 245
77	Computer Service Center 277
73	Computex Corp 235
71	Computhink 229
78	CT Micro Computer 58, 215, 233, 243
82	Covox 218
83	Creative Software 118
87	Cromemco 1, 2
72	Custom Computing 247
74	Cutting Edge of Technology 133
85	Cybernetics Inc 204
90	Cybertron 277
88	Cygol 231
91	Data Discount Center 201
94	Datamation 214
93	DataSearch 192
*	Data Set Cable Co 237
96	DDC Publications 241
89	Delta Products 271
84	Digital Engineering Inc 193
98	Digital Equipment Corp (Book Div.) 13
86	Digital Pathways 171
95	DRC (CA) 154
100	DRC (TX) 275
114	Eclectic (Div of Tano) 66
115	Electrolabs 280
120	Electronic Control Technology 239
125	Electronic Systems 278
126	Electronic Systems 279
130	Electronics Warehouse 287
132	Epoch Data Devices 247
133	Exidy 54, 55
134	Federal Communications Corp 123
135	Forethought Products 205
141	GRI 11
142	H Geller 194
143	General Scientific Corp 245
144	Gimix 247
150	Godbout 159
153	Go Forms 37
154	Gowan Industries 247
152	Graham Dorian 247
151	GRT 81
159	H & E Computronics 154
160	Hayden Book Co 212
161	Hexagon Systems 202
170	Hobby World 36, 255
174	Hollywood Systems 285
171	Houston Instruments 29
172	HUH 187
169	HUH 187
176	Independent Business Systems 213
177	Infinite Inc 247
173	Info 2000 151
178	Inmac 223
180	Intecolor (Div Intelligent Sys) 65
179	Integrand 189
175	Interactive Microware Inc 241
181	Internatinal Peripheral Systems 59
190	Ithaca Audio 281
195	Jade Co 272, 273
200	Jameco 256, 257
183	Key Supply Co 217, 231
203	Leedex Corp 235
*	Lifeboat Associates 120, 121
202	Mad Hatter Software 97
199	The Mail Mart 247
205	Marketline Systems 221
206	Marinchip Systems 103
207	Maxwell Data Products 105
*	McGraw Hill Book Co 25
215	Measurement Sys & Controls 173
216	Measurement Sys & Controls 231
213	Micro Ap 169
214	Micro Architect 245
217	MicroComputer Technology 147
220	MicroDaSys 189
219	Micro Diversions 35
221	Micro Focus LTD. 113
222	Micromail 72
204	Micro Mike's 216
*	Micropolis 190, 191
228	Micro Pro International 48
229	Microsette 277
*	Microsoft 195
232	Micro Software (CAI) 241
231	The Micro Works 206
233	Micro World 139
234	Miken Optical 245
230	Mikos 280
236	Mini Computer Suppliers 220
255	Morrow/Thinker Toys CIII
256	Morrow/Thinker Toys 39
257	Mountain Hardware 226
258	National Small Computer Show 107
281	NEECO 155
282	NEECO 165
280	Netronics 225
283	Newman Computer Exchange 263
285	North Star Computer 45
284	Ohio Micro Systems 109
*	onComputing 33
288	On Line 277
*	Oregon Software 217
292	Osborne & Associates 183
290	OSI CIV, 57, 89
*	Owens & Associates 203
295	Pacific Exchanges 231
296	Pacific Office Systems 284
297	PAIA 221
301	Per Com Data 116, 117
305	Per Com Data 253
302	Personal Software Inc 142, 143
*	Potomac Micro Magic 227
310	Potters Programs 277
312	Priority I 266, 267, 268
300	Processor Technology 94, 95
307	Program Design Inc 209
308	PRS 46,47
306	Quality Software 163
311	Quest Electronics 281
313	RACET Computes 192
314	Radio Shack Authorized Sales Center 277
322	RCA 85
323	Realty Software 67
*	The Recreational Programmer 245
325	RNB 233
326	RNB 243
316	S-100 Inc 203
317	Sara Tech Computers 210
331	S C Digital 237
318	Scelbi Computer Consulting Inc 219
*	Scientific Research 115
319	Michael Shrayer Software 185
*	Shugart 6, 7
320	Sierra Cybernetic Systems 215
327	Ed Smith's Software Works 194
328	Smoke Signal Broadcasting 73
*	Software Development & Training 215
340	Solid State Sales 265
350	Southwest Technical Products Corp CII
351	Speakeasy Software 131
335	SSM 77
352	Stirling/Bekdorf 124
353	Structured Systems Group 5
358	Sunflex Co Inc 245
354	Sunny International 285
357	SuperSoft 202
356	Sybex Inc 157
355	Synchro Sound 43
360	Tarbell Electronics 135
363	Technical Systems Consultants 137
364	Technology Systems South 237
365	Terrapin 235
266	Texas Instruments 31
366	Texas Instruments 179
368	3/M Company 91
371	Tora Systems Limited 231
372	Trans Data 204
374	TRS-80 Software Exchange 197
373	TSA Software 241
380	Ucatan 245
383	US Brokers Co (Div. CM Corp) 247
382	US Robotics 119
381	Vector Electronics 51
378	Vista 201
393	Vista 161
379	V M Professional Application Software 277
384	VR Data 282
387	Wameco 284
388	Western Digital Corp 32
389	Wintek 231
391	Worldwide Electronics 231
392	XComp 217
400	Xitex 239
401	Z_S Systems 247

**Correspond directly with Company*

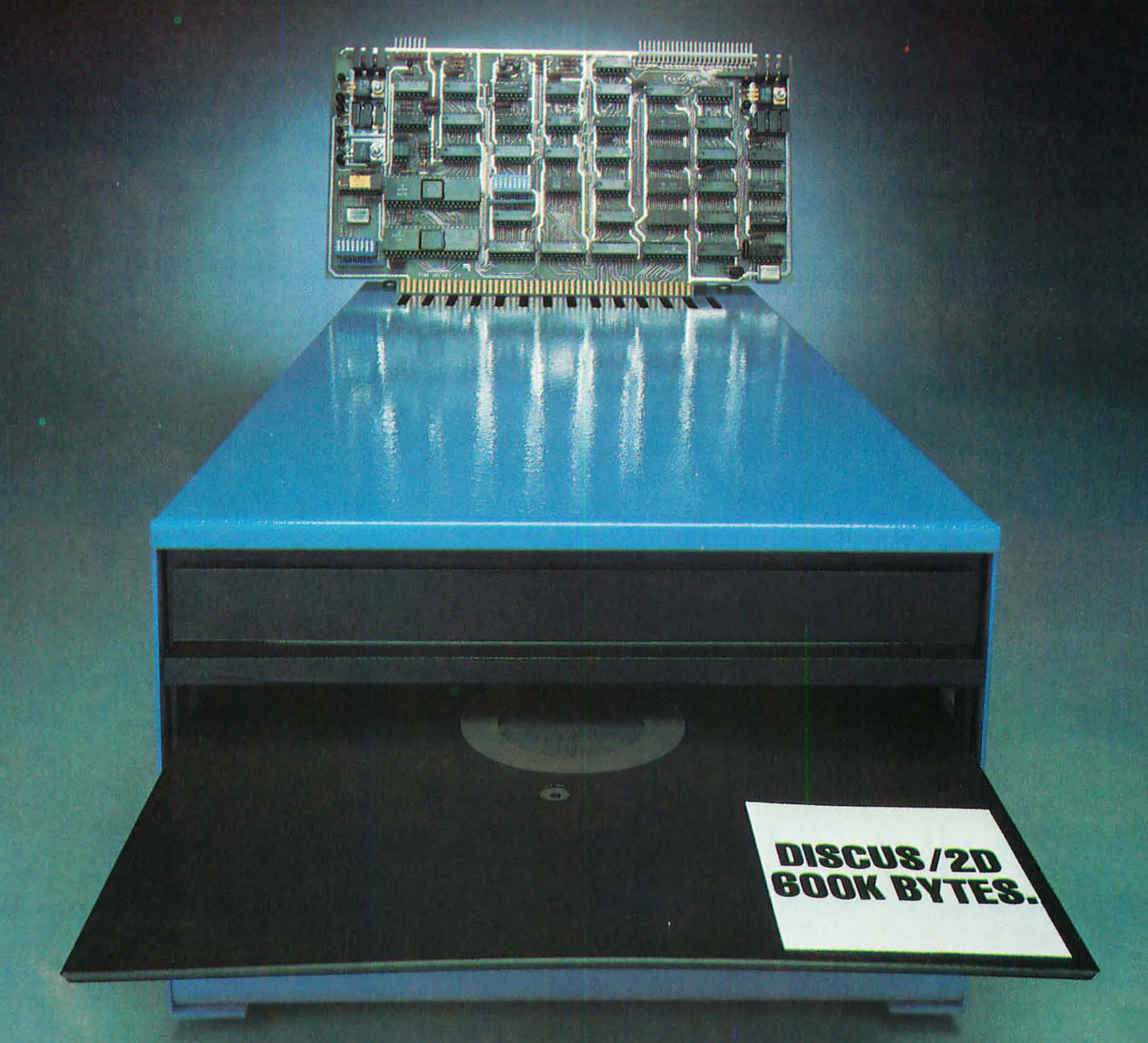
BOMB—BYTE's Ongoing Monitor Box

Article No.	ARTICLE	PAGE
1	Albus: A Model of the Brain for Robot Control	10
2	Allen: Simple Maze Traversal Algorithms	36
3	Ciarcia: Mind Over Matter	49
4	Watson: More Colors for Your Apple	60
5	Dawes: A Home for Your Computer	70
6	Gupton: Talk to a Turtle	74
7	Stanfield: My Computer Runs Mazes	86
8	Johnston: Computer Generated Maps	100
9	Powers: The Nature of Robots	132
10	Melton: The 1802 Op Codes	146
11	Reid-Green: History of Computing: The IBM 7070	148
12	Kiehn: Artificial Intelligence and Entropy	152
13	Ruckdeschel: BASIC Text Editor	156
14	Halsema: Bubble Memories	166
15	Radhakrishnan, Bhat: Stacks in Microprocessors	168
16	Van den Bout: Designing a Command Language	176
17	Linker: Timesharing: Squeezing the Most from Your Micro	228
18	Honess: Three Types of Pseudorandom Sequences	234

March BOMB Results

In the voting for the March 1979 BYTE, first prize and a $100 bonus check go to Ira Rampil for his article, "Preview of the Z-8000." Two articles shared second place, and will receive bonus checks of $50 each. These were the third installment of Joel Boney and Terry Ritter's article on the design and implementation of the Motorola 6809 processor, "A Microprocessor for the Revolution," and the second part of Andrew Filo's article, "Designing a Robot from Nature." Remember, it is your votes which determine whether an author will receive this bonus each month, so be sure to send in the BOMB evaluations.■

DOUBLE DENSITY



SOLID SAVINGS!

Now you can put your S-100 system solidly into a full-size, single/double density, 600K bytes/side disk memory for just $1149 complete.

DISCUS/2D™ single/double density disk memory from Thinker Toys™ is fully equipped, fully assembled, and fully guaranteed to perform perfectly.

DISCUS/2D™ is a second generation disk memory system that's compatible with the new IBM System 34 format. The disk drive is a full-size Shugart 800R, the standard of reliability and performance in disk drives. It's delivered in a handsome cabinet with built-in power supply.

The S-100 controller utilizes the amazing Western Digital 1791 dual-density controller chip . . . plus power-on jump circuitry, 1K of RAM, 1K of ROM with built-in monitor, and a hardware UART to make I/O interfacing a snap.

The DISCUS/2D™ system is fully integrated with innovations by designer/inventor George Morrow. Software includes BASIC-V™ virtual disk BASIC, DOS, and DISK-ATE™ assembler/editor. Patches for CP/M* are also included. CP/M*, MicroSoft Disk BASIC and FORTRAN are also available at extra cost.

DISCUS/2D™ is the really solid single/double density disk system you've been waiting for. We can deliver it now for just $1149. And for just $795 apiece, you can add up to 3 additional Shugart drives to your system. Both the hardware and software are ready when you are.

Ask your local computer store to order the DISCUS/2D™ for you. Or, if unavailable locally, write Thinker Toys,™ 5221 Central Ave., Richmond, CA 94804. Or call (415) 524-2101 weekdays, 10-5 Pacific Time. (FOB Berkeley. Cal. res. add tax.)

*CP/M is a trademark of Digital Research.

Morrow makes disk memory for

Thinker Toys™

Circle 255 on inquiry card.

"See us in booths 98 and 99 at the NCC"